

Chapitre 2 : Analyse et spécification des besoins

Introduction

La partie analyse et spécification des besoins est la partie fondamentale pour comprendre les fonctionnalités requises par le système développés. Il s'agit donc dans ce chapitre des objectifs de notre application, des acteurs du système, des besoins des utilisateurs que nous allons essayer de projeter dans des diagrammes de cas d'utilisation et des diagrammes de séquence accompagnés de description détaillée.

1. Analyse et spécification des besoins fonctionnels

1.1. Analyse des besoins fonctionnels

Nous analysons les différents besoins fonctionnels de notre application commençant par l'identification des acteurs.

1.1.1. Identification des acteurs

Un acteur représente un rôle d'un utilisateur qui interagit avec le système que nous modélisons. L'utilisateur peut être un utilisateur humain, une organisation, une machine ou un autre système externe. Notre plate-forme comporte les acteurs suivants :

- **Internaute** : C'est un visiteur non authentifié au site.
- **Utilisateur** : C'est un acteur abstrait. Tous les utilisateurs peuvent modifier leurs profils et commenter une ressource. Il existe trois types d'utilisateurs :
 - 1) **Éditeur** : Acteur qui s'est chargé principalement par la gestion des ressources.
 - 2) **Administrateur** : Administrateur de la plate-forme est un acteur occupé par la gestion des différents utilisateurs du site ainsi que par la gestion des différentes fonctionnalités offertes par la plate-forme. Un administrateur peut jouer le rôle d'un éditeur.
 - 3) **Client** : Acteur qui peut principalement acheter des ressources. Il existe deux types de clients dans notre plate-forme :

3.1 Individuel : Personne adhérent à la plate-forme, c'est un acteur qui crée son profil par soi-même. Il achète des ressources pour l'utilisation personnelle.

3.2 Organisation : Institution inscrite à la plate-forme. Elle achète des ressources pour les mettre à disposition de ses individus.

1.1.2. Expression des besoins fonctionnels

Les besoins fonctionnels qui seront exécutés par notre système sont :

❖ **Fonctions fournies aux internautes, à l'administrateur, aux éditeurs, aux individuels et aux organisations :**

- **S'inscrire** : Chaque internaute peut créer un compte dans la plate-forme en saisissant ses données personnelles dans une formulaire.
- **Rechercher ressource** : Tout acteur admet la possibilité de rechercher une ressource.
- **Consulter ressource** : Tout acteur peut consulter la page qui contient les informations d'une ressource.
- **Gérer panier virtuel** : Chaque acteur a le droit d'ajouter et supprimer des ressources de son panier virtuel. Il peut consulter son panier et valider ses commandes.
- **Choisir langage préféré** : Le visiteur du site peut appliquer une parmi trois langues à son affichage : anglais, français ou arabe.

❖ **Fonctions fournies aux utilisateurs (Administrateur, Éditeurs, Individuels et Organisations) :**

- **S'authentifier** : Un utilisateur peut s'authentifier en saisissant son email et son mot de passe, après avoir créé son compte. Il peut récupérer son mot de passe s'il l'a oubliée, en fournissant son email, il reçoit un lien de récupération de son mot de passe.
- **Modifier profil** : Tout utilisateur peut modifier son email et ses identifiants personnelles.
- **Commenter ressource** : Chaque utilisateur a le droit de rédiger des commentaires sur les ressources.

❖ **Fonctions fournies aux éditeurs et à l'administrateur :**

- **Gérer collections** : L'éditeur et l'administrateur peuvent ajouter des collections, consulter la liste des collections, modifier, chercher et supprimer des collections.

Chapitre 2. Analyse et spécification des besoins

- **Gérer ressources** : L'éditeur et l'administrateur peuvent ajouter des ressources (un éditeur doit ajouter au moins une collection pour qu'il puisse ajouter des ressources liées à cette collection). Ils peuvent consulter la liste des ressources, modifier, chercher, consulter ou supprimer les ressources de la liste. Ils ont aussi le droit de gérer le sommaire des ressources en ajoutant, modifiant et supprimant des chapitres.
- **Consulter statistiques** : L'éditeur et l'administrateur peuvent consulter les statistiques d'utilisation des ressources numériques.

❖ Fonctions fournies à l'administrateur :

- **Gérer utilisateurs** : Un administrateur admet la possibilité d'ajouter des utilisateurs en saisissant ses identifiants, il peut consulter la liste des utilisateurs, consulter le profil d'un utilisateur, modifier les informations d'un utilisateur, chercher un utilisateur et le supprimer.
- **Gérer auteurs** : Un administrateur a le droit d'ajouter un auteur, il peut consulter la liste des auteurs, consulter les informations d'un auteur, modifier, chercher et supprimer un auteur.
- **Gérer thèmes** : Un administrateur peut ajouter un thème, consulter la liste des thèmes, modifier, chercher et supprimer un thème.

❖ Fonctions fournies aux clients :

- **Acheter ressource** : Un client peut acheter une ressource, en ajoutant la ressource désirée au panier virtuel, consulter le panier et effectuer le paiement par une carte bancaire en remplissant le formulaire de paiement. Dès que le paiement s'est déroulé avec succès, l'utilisateur peut lire sa ressource en mode en ligne ou bien l'a téléchargée.
- **Gérer liste des intérêts** : Les clients peuvent ajouter et supprimer des ressources de leurs listes d'intérêts. Ils ont le droit de consulter les ressources de leurs listes.
- **Consulter liste d'achats** : En effectuant un achat, la ressource achetée se met automatiquement dans la liste d'achats. Un client peut consulter sa liste d'achats où il est apte de consulter et accéder ses ressources.

❖ Fonctions fournies aux individuels :

- **Créer profil** : Après qu'il s'inscrit et s'authentifie pour la première fois, le client individuel est demandé à saisir des informations spécifiques à son profil (date de naissance, profession, sexe...)

1.2. Diagramme de cas d'utilisation général

La figure 2.1 présente le diagramme de cas d'utilisation général qui permet d'identifier les grandes fonctionnalités du futur système et les possibilités d'interaction entre le système et les acteurs.

Chapitre 2. Analyse et spécification des besoins

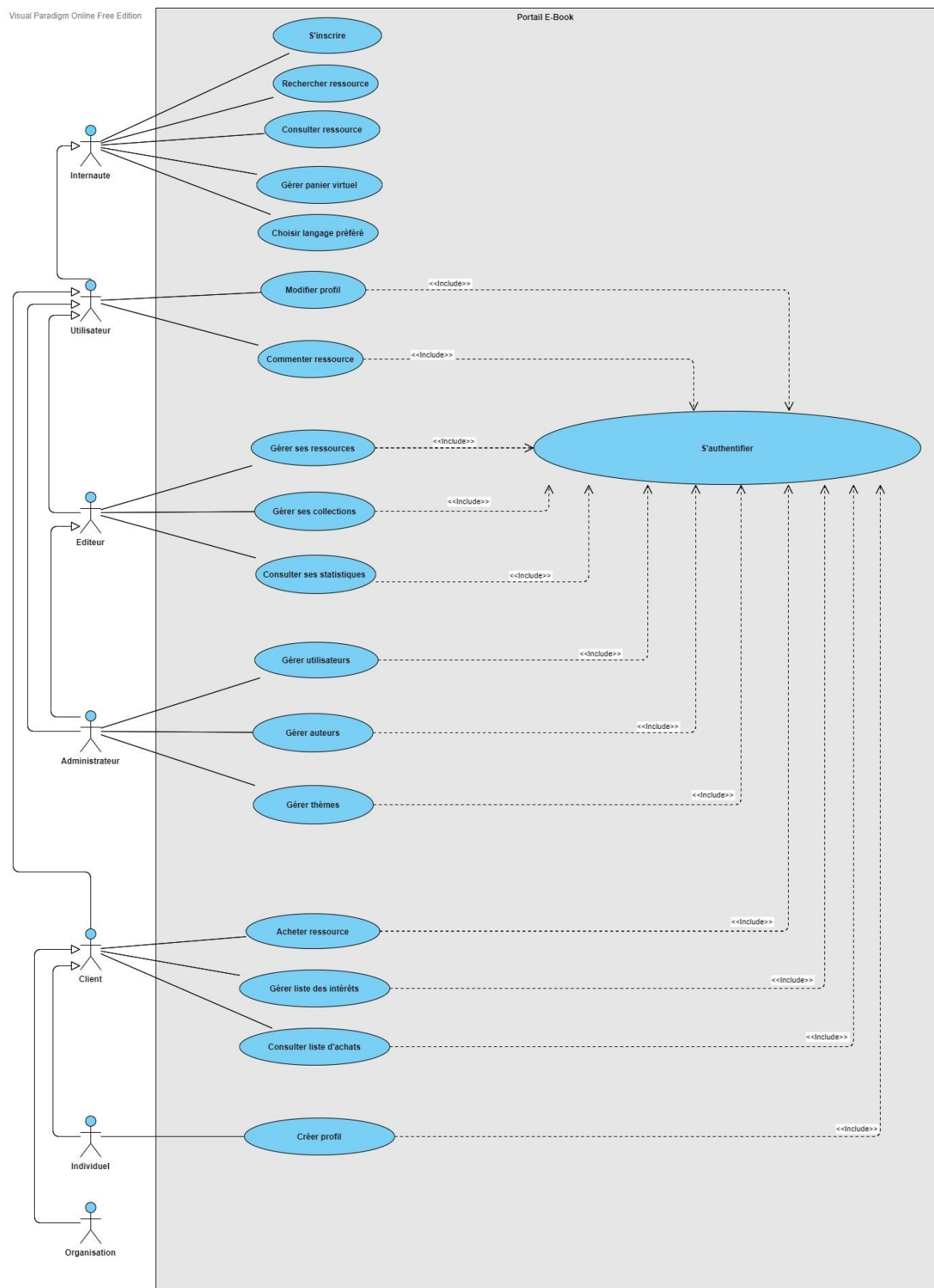


Figure 2.1 - Diagramme de cas d'utilisation général

1.3. Cas d'utilisation "Consulter ressource"

La figure 2.2 présente le diagramme de cas d'utilisation "Consulter ressource". Le visiteur du site admet le droit de consulter une ressource où il peut explorer les différentes informations sur la ressource (méta données, sommaire, abstrait, commentaire). Dans la section du méta données l'internaute peut consulter des informations sur l'auteur de la ressource en listant ses documents disponibles dans la plate-forme et il peut voir sa biographie. Le visiteur peut ajouter la ressource consultée à son panier et il peut l'a consultée en ligne ou bien l'a téléchargée dans le cas où la ressource est gratuite, mais il doit d'abord s'authentifier.

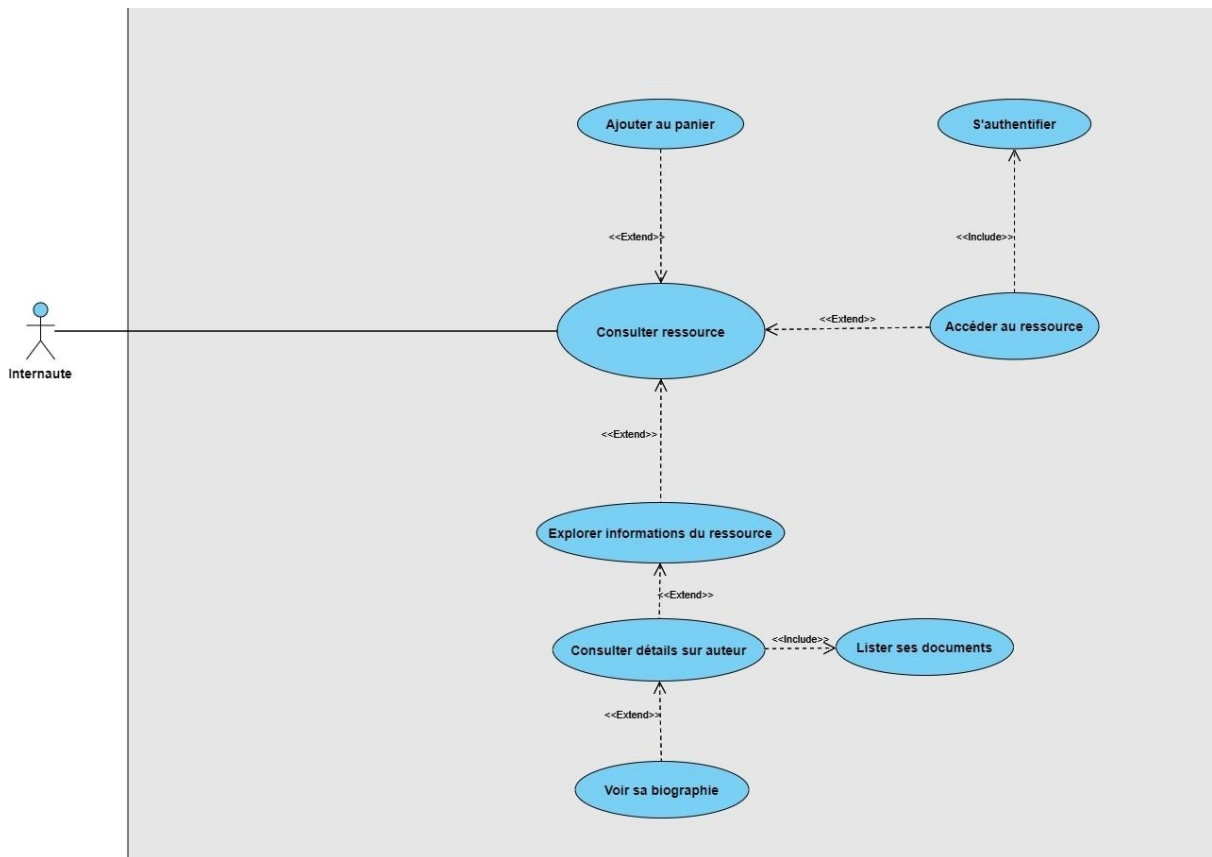


Figure 2.2 - Diagramme de cas d'utilisation : "Consulter ressource"

Dans la suite, la figure 2.3 présente le diagramme de séquence acteur/système "Consulter ressource" accompagnée d'une description textuelle structurée.

Diagramme de séquence "Consulter ressource"

Chapitre 2. Analyse et spécification des besoins

Visual Paradigm Online Free Edition

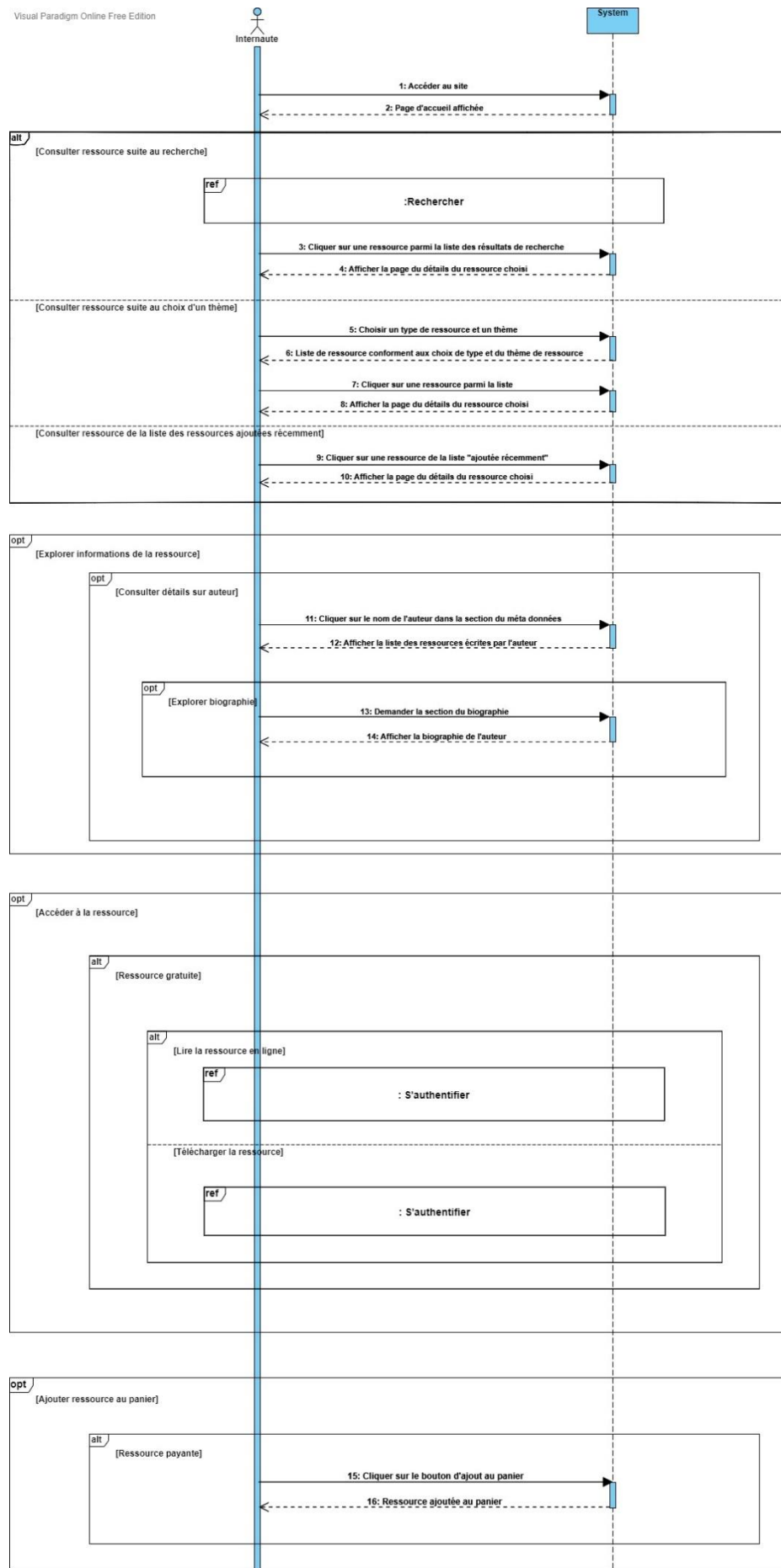


Figure 2.3 - Diagramme de séquence : "Consulter ressource"

Cas d'utilisation	Consulter ressource
Acteur	Internaute
Précondition	Ressource non consultée
Description du scénario nominal	<ul style="list-style-type: none"> - L'internaute accède au site - Le système lui affiche la page d'accueil - L'internaute consulte une ressource : <ol style="list-style-type: none"> 1. Consultation lors d'une requête de recherche 2. Consultation lors d'un choix du type et thème 3. Consultation de la liste "ajoutés récemment" -Le système affiche la page des informations du ressource choisit Extension : Explorer informations de l'auteur -L'internaute peut explorer des détails sur l'auteur en explorant les métas donnés -Le système liste les documents de l'auteur disponibles sur le site -L'internaute peut explorer la biographie de l'auteur -Le système affiche la biographie Extension : Accéder à la ressource - L'internaute peut l'accéder la ressource, si elle est gratuite : <ol style="list-style-type: none"> 1. Accéder la ressource en ligne, après qu'il s'authentifie 2. Télécharger la ressource, après qu'il s'authentifie Extension : Ajouter la ressource au panier -L'internaute peut ajouter la ressource au panier, si elle est payante -Le système ajoute la ressource au panier
Postcondition	Ressource consultée

Tableau 2.1 - Description du cas d'utilisation "Consulter ressource"

1.4. Cas d'utilisation "Gérer ses ressources"

La figure 2.4 présente le diagramme de cas d'utilisation "Gérer ses ressource". Un éditeur peut ajouter une ressource par 3 méthodes : Saisie manuelle des données bibliographiques, import d'un fichier MARC21 ou bien à l'aide du service Google Books API. Il peut consulter la liste de ses ressources ajoutées où il a droit de consulter ses ressources, modifier des informations de ses ressources ou supprimer ses ressources. Il existe dans sa liste des documents une zone de recherche qui lui permet de trouver une de ses ressources. Il peut

Chapitre 2. Analyse et spécification des besoins

ainsi gérer le sommaire de chaque document par un ajout, une modification ou une suppression des chapitres.

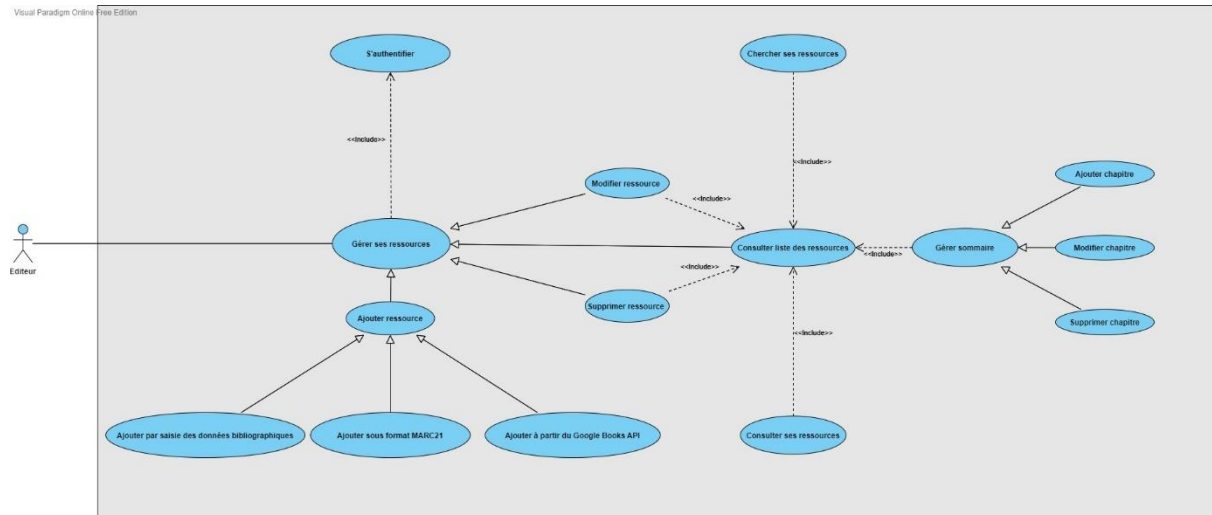


Figure 2.4 - Diagramme de cas d'utilisation : "Gérer ses ressources"

Dans la suite, la figure 2.5 présente le diagramme de séquence acteur/système "Ajouter ressource" accompagnée d'une description textuelle structurée.

Diagramme de séquence "Ajouter ressource"

Chapitre 2. Analyse et spécification des besoins

Visual Paradigm Online Free Edition

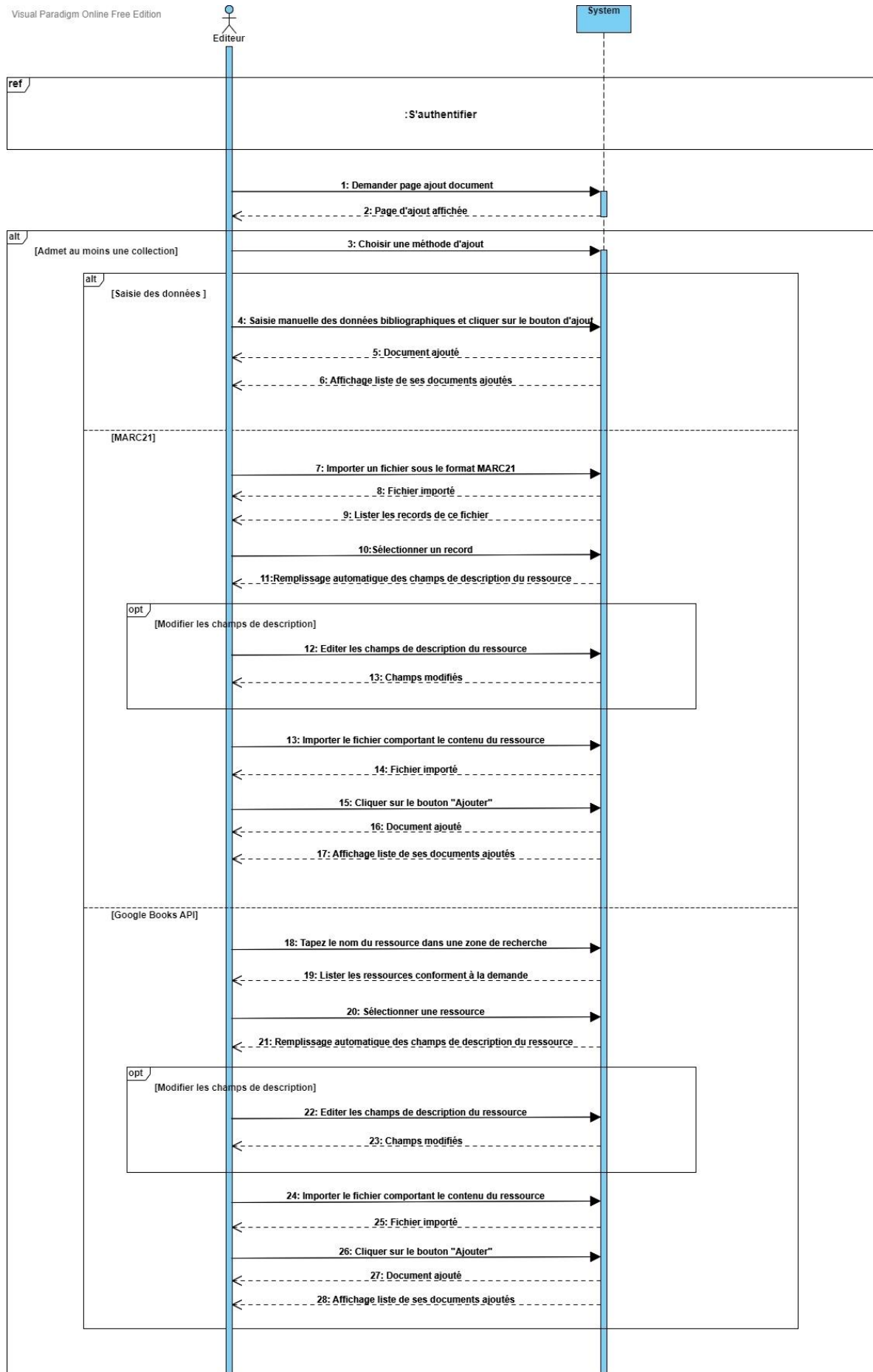


Figure 2.5 - Diagramme de séquence : "Ajouter ressource"

Cas d'utilisation	Ajouter ressource
Acteur	Editeur
Précondition	Editeur authentifié
Description du scénario nominal	<ul style="list-style-type: none"> - L'éditeur demande la page d'ajout du document - Le système lui affiche la page ajouter document - L'éditeur choisit une méthode pour ajouter son document : <ol style="list-style-type: none"> 1. Ajouter le document en saisissant manuellement ses données bibliographiques 2. Ajouter le document par un import d'un fichier MARC21 comportant les données bibliographiques du document à ajouter 3. Ajouter le document à l'aide du service Google Books API qui importe les données bibliographiques du document suite à une recherche sur le nom de la ressource -Le document est ajouté, le système affiche la liste des documents ajoutés par l'éditeur.
Postcondition	Ressource ajoutée
Exception	<ul style="list-style-type: none"> -S'il existe une erreur de saisie des champs, le système n'ajoute pas la ressource et renvoie une alerte de l'erreur. -Si l'éditeur n'a ajouté aucune collection, donc il ne peut pas ajouter des ressources sur la plate-forme.

Tableau 2.2 - Description du cas d'utilisation "Ajouter ressource"

1.5. Cas d'utilisation "Gérer utilisateurs"

La figure 2.6 présente le diagramme de cas d'utilisation "Gérer utilisateurs". Un administrateur peut ajouter des utilisateurs en saisissant ses identifiants. Il a le droit de consulter la liste des utilisateurs ou il existe une zone de recherche mise à sa disposition pour trouver un utilisateur à travers l'une de ses informations. Il peut consulter le profil d'un utilisateur, modifier ses informations, ou le supprimer.

Chapitre 2. Analyse et spécification des besoins

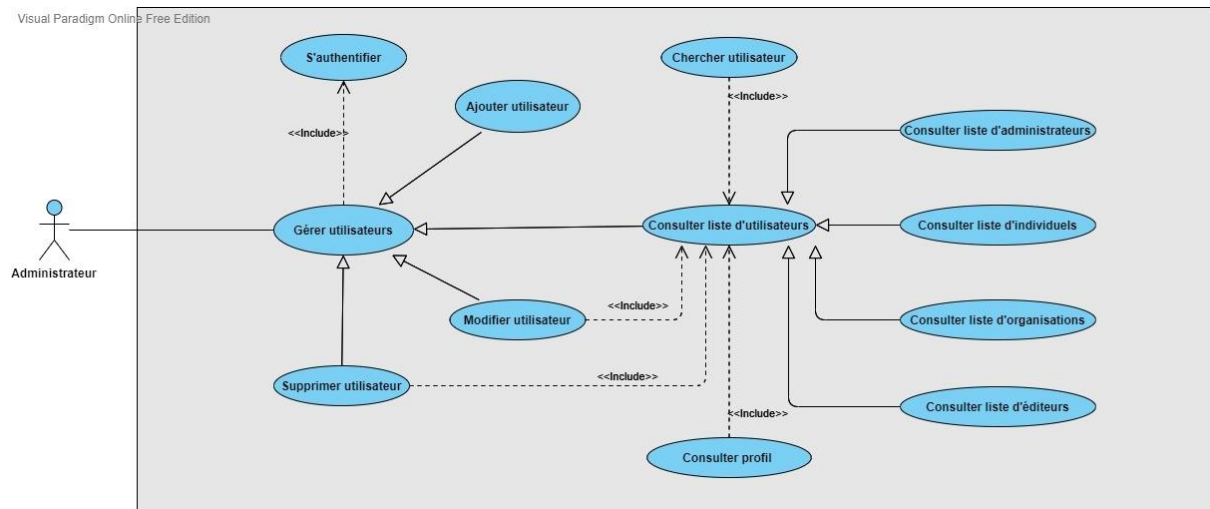


Figure 2.8 - Diagramme de cas d'utilisation : "Gérer utilisateurs"

Dans la suite, la figure 2.7 présente le diagramme de séquence acteur/système "Gérer utilisateurs" accompagnée d'une description textuelle structurée.

Diagramme de séquence "Gérer utilisateurs"

Chapitre 2. Analyse et spécification des besoins

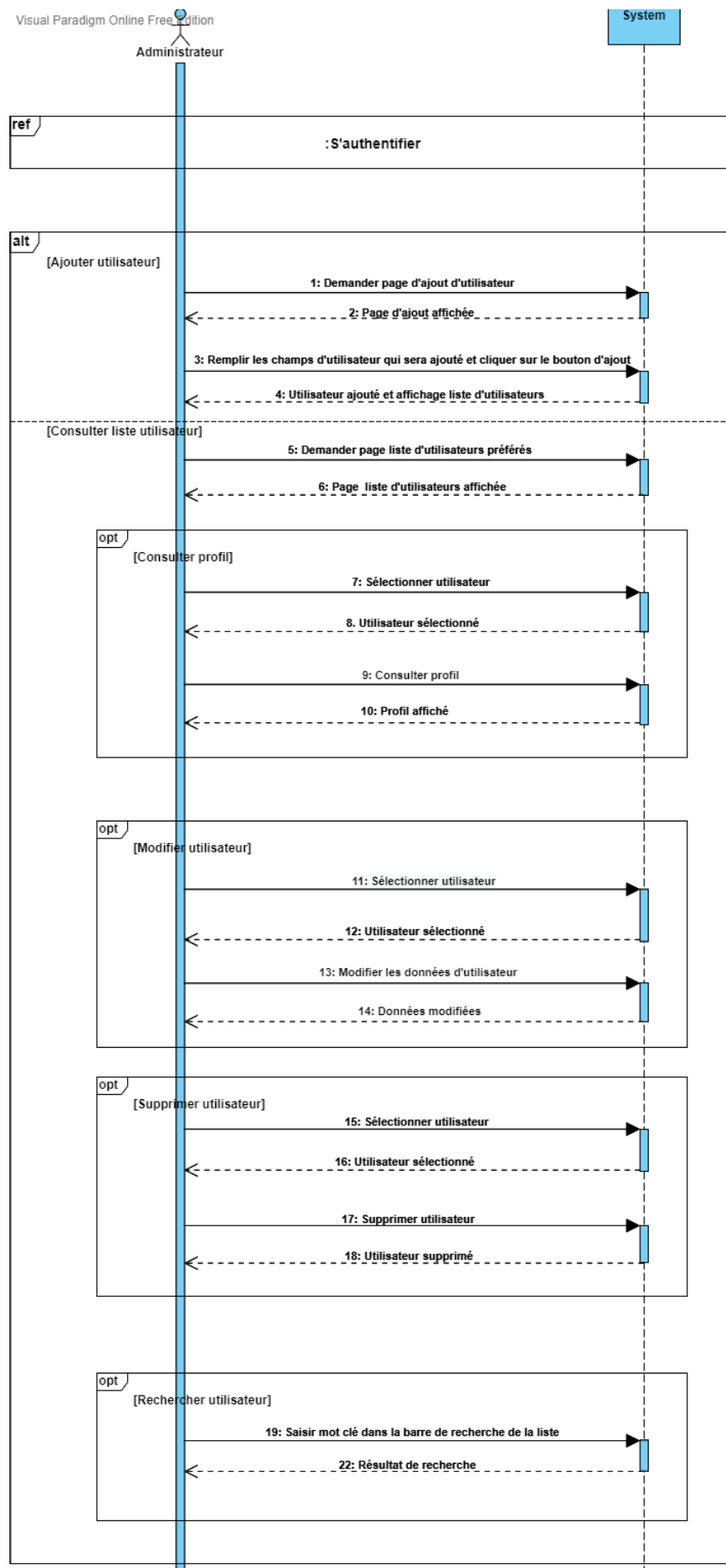


Figure 2.7 - Diagramme de séquence : "Gérer utilisateurs"

Cas d'utilisation	Gérer utilisateurs
Acteur	Administrateur
Précondition	Administrateur authentifié
Description du scénario nominal	<p>1. L'administrateur ajoute un utilisateur</p> <ul style="list-style-type: none"> - Administrateur demande d'ajouter un utilisateur - Le système affiche la page ajouter utilisateur - L'administrateur saisit les coordonnées de l'utilisateur et valide ses saisis. - L'utilisateur est ajouté, le système affiche la liste d'utilisateurs ayant le même type. <p>2. L'administrateur consulte la liste des utilisateurs</p> <ul style="list-style-type: none"> - Administrateur choisit une des 4 listes d'utilisateurs - Le système affiche la liste désirée <p>Extension 1 :</p> <ul style="list-style-type: none"> - Administrateur consulte le profil d'un utilisateur <p>Extension 2 :</p> <ul style="list-style-type: none"> - Administrateur modifie les informations d'utilisateur <p>Extension 3 :</p> <ul style="list-style-type: none"> - Administrateur supprime un utilisateur <p>Extension 4 :</p> <ul style="list-style-type: none"> - Administrateur cherche un utilisateur
Postcondition	<ul style="list-style-type: none"> - Utilisateur ajouté - Liste d'utilisateur consultée - Profil utilisateur consulté - Utilisateur modifié - Utilisateur recherché - Utilisateur supprimé
Exception	-S'il existe une erreur de saisie, le système renvoie une alerte de d'erreur.

Tableau 2.3 - Description du cas d'utilisation "Gérer utilisateurs"

1.6. Cas d'utilisation "Acheter ressource"

La figure 2.8 présente le diagramme de cas d'utilisation "Acheter ressource". Un client peut acheter une ressource. Il ajoute la ressource au panier, consulte ce dernier puis effectuer le paiement pour terminer l'opération d'achat. Après qu'il consulte sa ressource depuis sa liste d'achat, il a le droit lire la ressource en ligne ou bien il peut l'a téléchargée

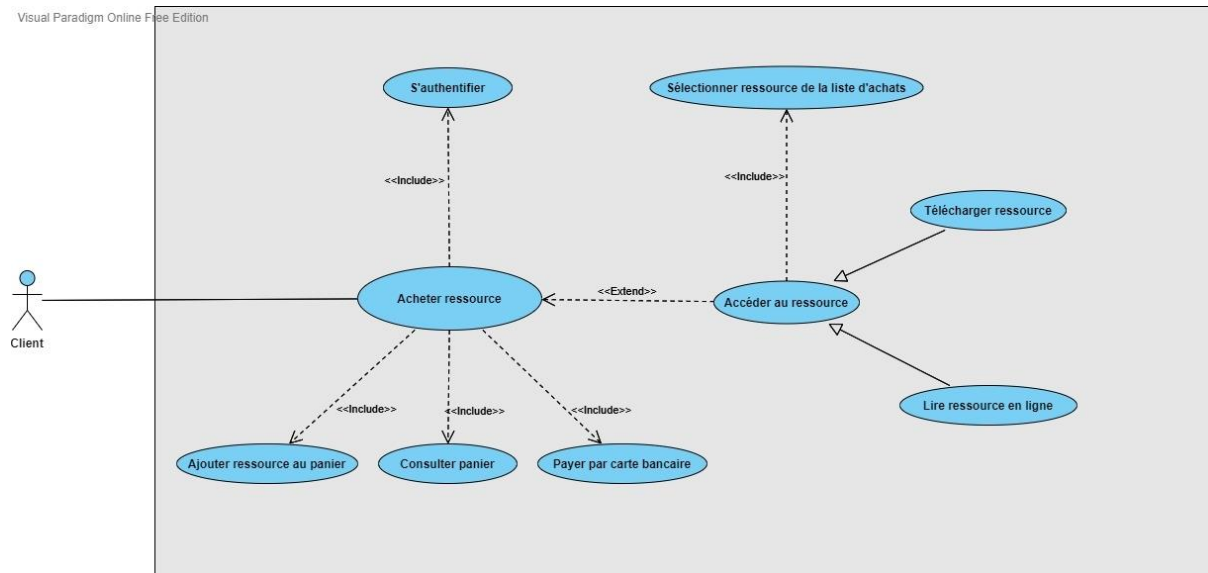


Figure 2.8 - Diagramme de cas d'utilisation : "Acheter ressource"

Dans la suite, la figure 2.9 présente le diagramme de séquence acteur/système "Acheter ressource" accompagnée d'une description textuelle structurée.

Diagramme de séquence "Acheter ressource"

Chapitre 2. Analyse et spécification des besoins

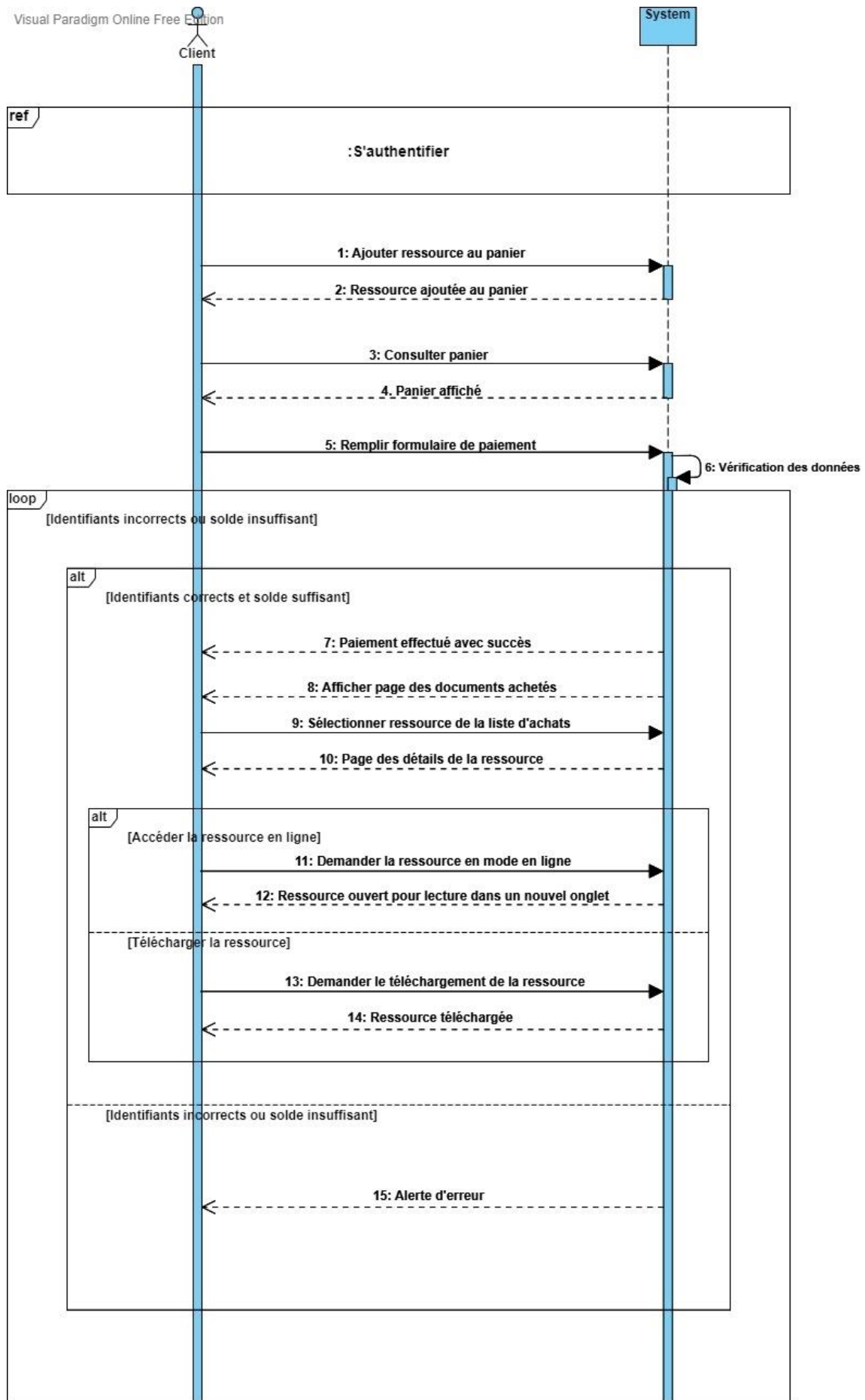


Figure 2.9 - Diagramme de séquence : "Acheter ressource"

Cas d'utilisation	Acheter ressource
Acteur	Client
Précondition	Client authentifié
Description du scénario nominal	<ul style="list-style-type: none">- Le client ajoute une ressource au panier- Le client consulte son panier- Le système affiche la page du panier- Le client remplit le formulaire de paiement- Le paiement est réalisé avec succès, le système affiche la page liste d'achats- Le client sélectionne une ressource qu'il veut l'accédée- Le système affiche la page des détails de la ressource- Le client choisit la façon d'accéder à sa ressource : <ol style="list-style-type: none">1) Demander la ressource en mode en ligne2) Télécharger la ressource
Postcondition	Ressource achetée
Exception	<ul style="list-style-type: none">- S'il existe une erreur de saisie, le système renvoie une alerte de d'erreur.- S'il existe une erreur de paiement, le système affiche un message d'erreur

Tableau 2.4 - Description du cas d'utilisation "Acheter ressource"

2. Analyse et spécification des besoins techniques

2.1. Choix technique

L'organisme d'accueil a exigé d'utiliser le framework ASP.NET Core 3.1 côté back-end. Dans la suite nous allons présenter certains avantages de ce framework.

2.1.1. Framework ASP.NET Core

ASP.NET Core est un Framework Web gratuit et open-source, développé par Microsoft. Ce framework s'exécute sur macOS, Linux et Windows. ASP.NET Core a été publié pour la première fois en 2016 et est une refonte des versions antérieures d'ASP.NET réservées uniquement à Windows. Il contient un framework MVC pour la création d'applications Web et d'API à l'aide de modèle de conception modèle-vue-contrôleur.

On a considéré l'utilisation de d'ASP.NET Core comme un choix technique pour les raisons suivantes :

- **Moderne et innovatif** : ASP.NET Core est conçu pour permettre aux composants d'exécution, aux API, aux compilateurs et langages d'évoluer rapidement, tout en fournissant une plate-forme stable et prise en charge pour maintenir les applications en cours d'exécution.
- **Open-source** : ASP.NET Core est open source sur GitHub. Il est possible à chacun de télécharger les sources, compiler soit même le framework et surtout contribuer.
- **Déploiement flexible** : Le runtime ASP.NET Core sur lequel l'application s'exécute peut-être déployer dans le cadre de l'application ou installé de manière centralisée sur le serveur web.
- **Testabilité** : Les tests unitaires sont très faciles dans les applications de base .Net Core.

2.2. Spécification des besoins non fonctionnels

Afin d'assurer un bon fonctionnement de l'application et de garantir la satisfaction de l'utilisateur, des contraintes doivent être prises en compte tout au long du développement du projet :

- **Performance et stabilité** : Nous considérons à rendre le système stable et performant. Il doit être en mesure d'exécuter ses fonctions pour une période de temps spécifique ainsi que le temps de réponse et le chargement de l'application, ouverture d'écran et des délais de rafraichissement.
- **Ergonomie et convivialité** : L'application doit fournir une interface simple et élégante pour l'utilisateur afin de faciliter l'exploitation des services de l'application.
- **Maintenance et évolution** : Le code doit être bien lisible, compréhensible et modulaire pour garantir la souplesse, l'évolution et la maintenance de la solution. Le site web devra donc être facilement évolutif pour pouvoir implémenter très rapidement de nouvelles fonctionnalités importantes et ainsi répondre aux changements du marché.
- **Sécurité** : De nos jours, les applications web sont de plus en plus exposées au danger des pirates, donc assurer la sécurité d'une application web est devenu quelque chose d'obligatoire pour protéger le serveur, les clients et leurs données.

Conclusion

Dans ce chapitre, on a identifié les besoins fonctionnels et non-fonctionnels. Nous avons détaillé les besoins fonctionnels via des diagrammes de cas d'utilisation et de séquence

Chapitre 2. Analyse et spécification des besoins

acteur système. Nous avons aussi présenté les besoins techniques que nous considérons. Le chapitre suivant est dédié à la conception de notre plate-forme.