



Université Abdelhamid Mehri – Constantine 2
2020/2021. Semestre 5

Génie Logiciel 2

– Cours 2 –

Chapitre 2 : Étude préliminaire et spécification des besoins



Staff pédagogique			
Nom	Grade	Faculté/Institut	Adresse e-mail
Dr. Sahar SMAALI	MCB	Nouvelles technologies	sahar.smaali@univ-constantine2.dz

Étudiants concernés			
Faculté/Institut	Département	Niveau	spécialité
Nouvelles technologies	TLSI	Licence 3	GL

Objectifs du cours 2

-
-
-

Sommaire

1	Introduction	2
2	Exemple: Système de gestion d'un hôpital	2
3	Étude préliminaire	2
3.1	Analyse du domaine	2
3.2	Définition de la problématique et du cadre du projet	3
3.3	Recueil des besoins	4
4	Spécification des besoins	5
4.1	Identification des acteurs	6
4.2	Identification des cas d'utilisation	6
4.3	Structuration des cas d'utilisation en packages	7
4.4	Identification des relations	8
4.5	Description textuelle des Cas d'utilisation	9
4.6	Exemple d'une fiche descriptive	10
4.7	Description graphique des cas d'utilisation	10
5	Classement des cas d'utilisation et planification des itérations	11
6	Production des maquettes IHM	11
7	Conclusion	12

1 Introduction

Le processus unifié est un processus de développement logiciel, c'est-à-dire qu'il regroupe les activités à mener pour transformer les besoins d'un utilisateur en système logiciel. Ce chapitre explique la première activité de ce processus qui comprend l'étude préliminaire du projet et le recueil des besoins ainsi que leur spécification avec les diagrammes UML correspondants. Un exemple de système de gestion des dossiers des patients hospitalisés dans un hôpital est exploité pour illustrer ces activités tout au long de ce chapitre.

2 Exemple: Système de gestion d'un hôpital

On souhaite développer un système d'information pour gérer les dossiers médicaux des patients hospitalisés dans un hôpital. L'hôpital est structuré en services (administration, pédiatrie, chirurgie...). A chaque service sont affectés des médecins spécialistes, des infirmières et des aides-soignantes. Chaque service a un chef de service qui est aussi un médecin. Chaque service contient des chambres et des lits. Le dossier médical du patient doit contenir des informations sur son identité, ainsi que sur toutes ses hospitalisations et les soins qu'il a reçus. Le dossier médical du patient doit contenir aussi les hospitalisations effectuées dans d'autres hôpitaux.

Le dossier médical est édité par une secrétaire médicale lorsque le patient est admis pour hospitalisation dans un service. Elle enregistre les informations sur l'admission du patient (genre : service d'hospitalisation, médecin traitant, date d'entrée et date de sortie, etc.) et lui attribue une chambre libre dans le service d'hospitalisation. Si le patient possède déjà un dossier médical dans l'hôpital, la secrétaire médicale doit juste mettre à jour les informations de ce patient (genre : nouvelle mutuelle, changement d'adresse ou de téléphone, etc.). Si le patient est admis pour la première fois, elle doit créer son dossier. Elle enregistre toutes les informations du patient : son numéro de sécurité sociale, adresse, téléphone, mutuelle, personne à contacter, etc...

A chaque visite d'un médecin, une infirmière doit saisir dans le dossier médical un compte rendu de la visite. Ce compte rendu pourra contenir une prescription et une demande de soin. A chaque fois que le patient reçoit des soins, une infirmière doit les enregistrer dans le dossier. Vu la sensibilité des informations contenues dans le dossier médical, l'authentification par login et mot de passe des utilisateurs de l'application est indispensable.

3 Étude préliminaire

L'étude préliminaire est la toute première étape du processus de développement UP. Elle consiste à effectuer un premier repérage des besoins fonctionnels et opérationnels, en utilisant principalement le texte, ou des diagrammes très simples. Au début d'un projet à réaliser, l'important est de définir de quoi les utilisateurs ont vraiment besoin, dans quel contexte, et ce qu'ils comptent en faire. La première opération consiste alors à se doter d'une vision générale (Quoi + Pourquoi + Combien) du produit afin de le mieux caractériser et de dégager des éléments d'appréciation. Cette vision est généralement exprimée dans un cahier des charges préliminaire.

3.1 Analyse du domaine

L'analyse de domaine permet aux développeurs d'apprendre suffisamment d'informations pour pouvoir comprendre le problème et prendre les bonnes décisions lors de l'analyse des exigences et des autres étapes du processus. Elle consiste en la collection des informations à partir de toute source disponible en incluant les experts du domaine, les livres, les sites Web, les logiciels existants et leurs documentations, etc.

Document de l'analyse du domaine

Les informations collectées lors de l'analyse du domaine doivent être réunies dans un document qui sera diffuser aux différents membres de l'équipe de développement. Ce document résume la motivation de l'analyse du domaine en précisant le problème à résoudre par le développement ou l'extension d'un système et il contient au minimum les sections suivantes:

- **Glossaire:** Il liste les signification de tous les termes utilisés dans le domaine qui ne font pas partie du langage courant ou qui ont une signification particulière.
- **Connaissances générales du domaine:** Elle résume les principes scientifiques et les techniques d'analyse (largement connus des experts du domaine) qui sont indispensables pour acquérir une compréhension du problème et sa solution.
- **Étude de l'existant:** Les systèmes disponibles sur le marché pour aider les utilisateurs et les clients à résoudre le problème.

Exemple d'un document d'analyse de domaine

- **Introduction** Le directeur d'un hôpital nous demande de lui développer un système d'information pour la gestion des dossiers de l'hôpital. Ce document résume les informations que nous devons apprendre sur ce domaine.
- **Glossaire** Expliquer les termes et abréviations utilisés dans les dossiers de l'hôpital par les médecins et les infirmières.
- **Connaissance** Cette section doit fournir l'organigramme de l'hôpital ainsi que des statistiques sur les patients, les soins et les services offerts ainsi que le staff médical et administratif.
- **Clients et utilisateurs** Tout les employés de l'hôpital : les médecins, les infirmières, les secrétaires et les directeurs.
- **Étude de l'existant** Analyser le système ancien de l'hôpital et étudier la possibilité de l'améliorer. Analyser les systèmes existants sur le marché.

Avantages de l'analyse du domaine

- Elle offre une communication plus efficace avec les clients et les utilisateurs.
- Elle offre une meilleure compréhension des problématiques des clients.
- Elle permet d'assurer que les solutions adoptées résolvent les problèmes des clients.

3.2 Définition de la problématique et du cadre du projet

Une fois l'analyse du domaine terminée, il est nécessaire de délimiter le projet et cerner sa problématique avant d'approfondir l'analyse des exigences détaillées. La définition claire de la problématique et la portée du projet permet d'évaluer les résultats et d'éviter de concevoir des solutions non adéquates ou inutiles.



Remarque

Une problématique représente une difficulté rencontrée par les utilisateurs ou les clients, ou une opportunité qui se traduira par des avantages tels qu'une amélioration de la productivité ou des ventes.

Le problème à résoudre doit être déclaré d'une manière simple, courte et succincte. Si le problème est compliqué et composé d'une longue liste de sous-problèmes, il est préférable de restreindre le cadre du projet en définissant une problématique plus précise et en excluant certains de ces sous-problèmes

(qu'ils peuvent être laissés pour un autre projet). En d'autres termes, il s'agit de définir le périmètre du système en précisant ce qui fait partie du système.

Exemple

- **Problématique:** Comment automatiser et faciliter la gestion des dossiers de l'hôpital?
Pour cerner la problématique et limiter la taille du système, on ne s'intéresse dans ce cours uniquement à la gestion des dossier médicaux des patients hospitalisés.
- **Objectifs du projet :**
 - Éviter toute perte d'informations des patients
 - Gagner du temps
 - Faciliter la gestion des chambres des services
 - Etc.

3.3 Recueil des besoins

Le recueil des besoins est l'une des étapes les plus importantes dans la réalisation d'un projet de développement d'un système logiciel. En effet, elle est déterminante pour la suite et sert à la définition du cahier de charges. L'échec de cette activité implique l'échec total du projet.

Le recueil des besoins consiste à découvrir ce que les utilisateurs ont besoin que le système fasse. Un besoin est une caractéristique que le système doit avoir ou une contrainte qu'il doit satisfaire pour être accepté par le client et par les utilisateurs. Les besoins sont classés en deux catégories : fonctionnels et non-fonctionnels.

- **Besoins fonctionnels** définissent le comportement de base du système et précisent ce que le système doit faire. En d'autres termes, ils spécifient une fonction ou un service que le système doit offrir à ses utilisateurs tels que les actions d'ajout, de modification, et d'annulation.
- **Besoins non-fonctionnels** décrivent les contraintes imposées au système pour améliorer son fonctionnement tels que les choix techniques et les besoins de fiabilité, performance, sécurité et utilisabilité.

Étant donné que les utilisateurs sont la principale source des besoins, cette activité se concentre uniquement sur la vue de l'utilisateur du système. Autres aspects tels que la structure du système ou la technologie choisie pour le construire et ceux non directement visibles par l'utilisateur ne font pas partie des besoins recueillis.

Sources de recueil des besoins

Les besoins peuvent être recensés à partir de diverses sources à savoir :

- les utilisateurs directs,
- les personnes ayant une relation avec le système,
- autres systèmes avec lequel va interagir le système,
- le matériel sur lequel sera déployé le système,
- les contraintes juridiques et administratives,
- les contraintes techniques.

Techniques de recueil des besoins

Plusieurs moyens sont utilisés pour recenser les besoins à savoir:

- Analyse des systèmes et documents actuels,
- Interviews avec les clients et les utilisateurs,
- Questionnaires aux utilisateurs,
- Ateliers de brainstorming.

La gestion des besoins se fait de manière itérative et incrémentale, il ne s'agit pas de définir et de stabiliser les besoins une fois pour toutes dès la première phase du projet avant de commencer à développer, mais dans un contexte où les souhaits des intéressés peuvent ne pas être dès le départ explicites et donc changent inévitablement.

Exemple

Le système de l'hôpital à concevoir comprend différents types de fonctionnalités telles que: la gestion du personnels, des patients, des équipements et du matériels. Vu la grande taille du système, On ne s'intéresse dans la suite de ce cours uniquement à la gestion des dossiers médicaux des patients hospitalisés. La spécification des autres besoins peut d'être retardée à d'autre cycle de développement. Le tableau ci-dessous donne un exemple des besoins recueillis lors du premier cycle de développement du système.

Besoins Fonctionnels	Besoins non-fonctionnels
<ul style="list-style-type: none"> ● Admission d'un patient ● Création des dossiers des patients ● Enregistrement des comptes rendu des patients 	<ul style="list-style-type: none"> ● Authentifications des secrétaires et des infirmiers ● Création des comptes utilisateurs ● Le dossier médical du patient doit contenir aussi les hospitalisations effectuées dans d'autres hôpitaux.

4 Spécification des besoins

Un système est destiné à être utilisé par des utilisateurs (humains ou machines). Le point de vue des utilisateurs est important pour comprendre le fonctionnement du système et les services attendus. La spécification des besoins permet de :

- Identifier des fonctionnalités communes entre cas d'utilisation pour en déduire des cas d'utilisation plus précises qui seront utilisées par inclusion, extension ou généralisation/spécialisation.
- Identifier des interactions de bas niveau.
- Éliminer des inconsistances potentielles.
- Valider les cas d'utilisation avec le client.
- Faciliter la compréhension des besoins et du système par toutes les parties prenantes.
- Optimiser le diagramme des cas d'utilisation.

Le modèle des cas d'utilisations permet de décrire, d'un point de vue utilisateur, le système étudié. La figure 1 illustre la démarche à suivre pour élaborer ce modèle. L'identification des acteurs et cas d'utilisation se fait lors de la phase d'Inception de UP. Il s'agit d'un travail préliminaire visant à définir le périmètre fonctionnel du projet. Cependant, la description détaillée des cas d'utilisation (textuelle et graphique) des cas d'utilisation se fait principalement lors de la phase d'Élaboration, mais de façon itérative et incrémentale.

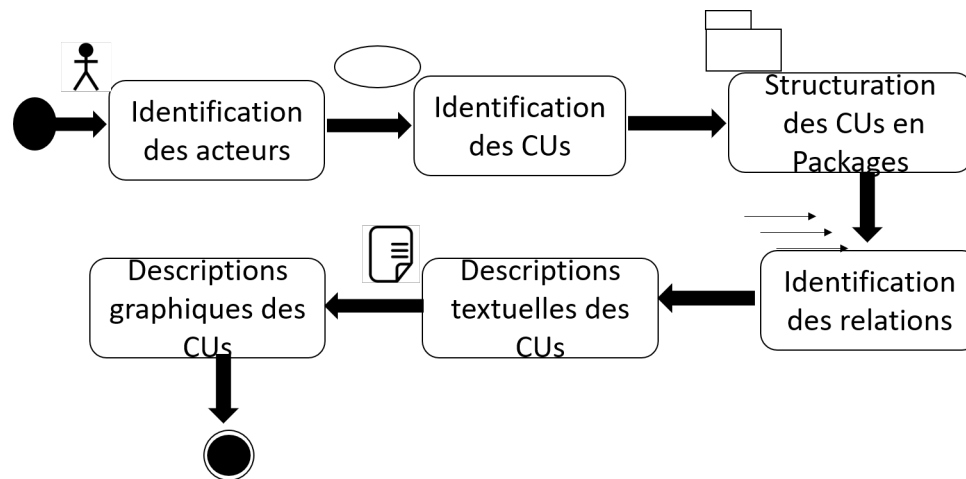


Figure 1: Démarche d'élaboration du modèle des cas d'utilisations .

4.1 Identification des acteurs

Un acteur représente un rôle joué par des entités externes (utilisateurs humains, dispositifs matériels ou autres systèmes) qui interagissent directement avec le système étudié. Un acteur peut consulter et/ou modifier directement l'état du système, en échangeant des messages. Il ne faut pas confondre la notion d'acteur avec la notion d'utilisateur. Un acteur décrit le rôle qu'un utilisateur joue par rapport au système. Un utilisateur peut jouer successivement différents rôles par rapport au système, et être modélisée par plusieurs acteurs. On distingue 2 types d'acteurs :

- des acteurs principaux ou primaires qui utilisent le système pour accomplir leurs buts.
- des acteurs secondaires qui peuvent fournir ou recevoir de l'information, ou s'occuper de la supervision ou de l'entretien du système.

Afin d'identifier les acteurs, on doit répondre aux questions suivantes :

- Qui ou quoi utilise le système?
- Qui ou quoi interagit avec le système?
- Qui contrôle ou maintient le système?

Exemple

A partir de l'étude préliminaire et les discussion effectuées avec le client, les acteurs suivants sont identifiés et résumés dans le tableau ci-dessous:

Acteur	Type	Rôle
Secrétaire	Principal	représente le personnel administratif responsable sur la gestion des chambres et de lits dans l'hôpital.
Infirmière	Principal	représente le personnel responsable sur la gestion des dossiers médicaux des patients.

4.2 Identification des cas d'utilisation

Un cas d'utilisation représente une fonctionnalité ou un service fournie par le système. Il est spécifié par une séquence d'actions qui produisent un résultat observable intéressant pour un acteur particulier. Les cas d'utilisations offrent une abstraction du dialogue entre les acteurs et le système en décrivent les interactions potentielles entre eux sans entrer dans les détails.

Pour identifier les cas d'utilisation, il faut répondre aux questions

- Comment chaque acteur utilise-t-il le système?

- Que fait le système pour chaque acteur?
- Quelles sont les fonctionnalités du système attendues par chaque acteur?

Exemple

Le tableau ci-dessous récapitule les cas d'utilisation attribués à chaque acteur identifié.

Acteur	cas d'utilisation
Secrétaire	<ul style="list-style-type: none"> ● Admettre un patient ● Créer le dossier médical d'un patient ● Mettre à jour le dossier médical d'un patient ● S'authentifier
Infirmière	<ul style="list-style-type: none"> ● Saisir compte rendu d'une consultation d'un patient. ● Enregistrer les soins offerts à un patient. ● S'authentifier

4.3 Structuration des cas d'utilisation en packages

Le diagramme des cas d'utilisation d'un système peut très vite devenir encombrant et difficile à décortiquer. De ce fait, il est conseillé de regrouper les cas d'utilisation par acteur ou par domaine fonctionnel en paquetages. Le diagramme de paquetage (package) permet d'avoir des ensembles cohérents de cas d'utilisation et de découper le problème en sous-problèmes (sous-systèmes).

Exemple

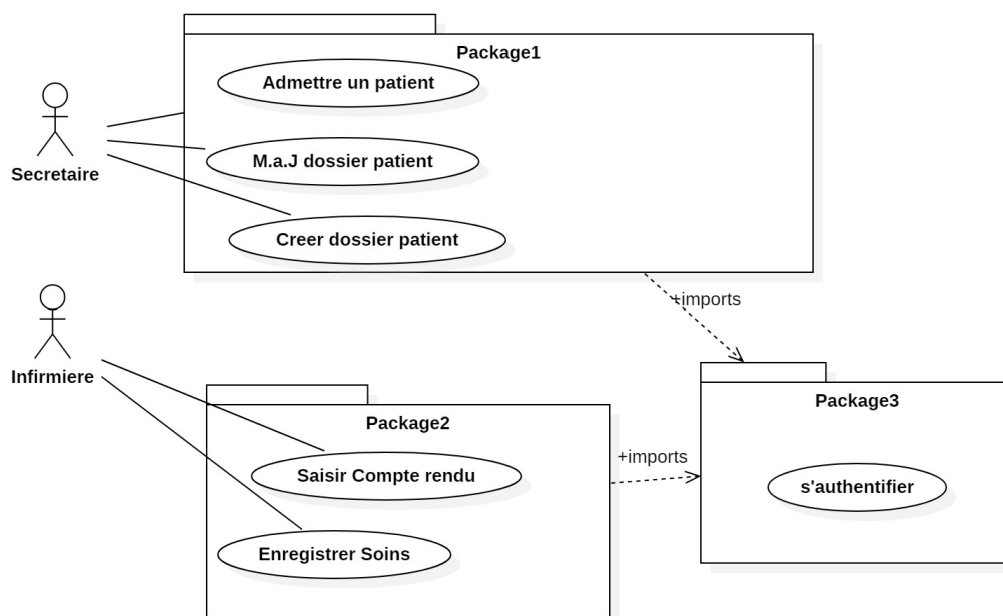


Figure 2: Structuration en packages en fonction des acteurs.

4.4 Identification des relations

Pour affiner le diagramme de cas d'utilisation, UML définit trois types de relations standardisées entre cas d'utilisation :

- L'inclusion signifie qu'un cas d'utilisation en incorpore explicitement un autre cas de façon obligatoire. Elle permet d'enlever la redondance dans le modèle.
- L'extension signifie qu'un cas d'utilisation en incorpore implicitement un autre, de façon optionnelle. Elle permet de séparer le comportement optionnel ou rare du comportement obligatoire.
- La généralisation/spécialisation : permet d'exprimer un comportement commun entre plusieurs cas d'utilisation.

La seule relation qui peut exister entre les acteurs est celle de généralisation.

Exemple

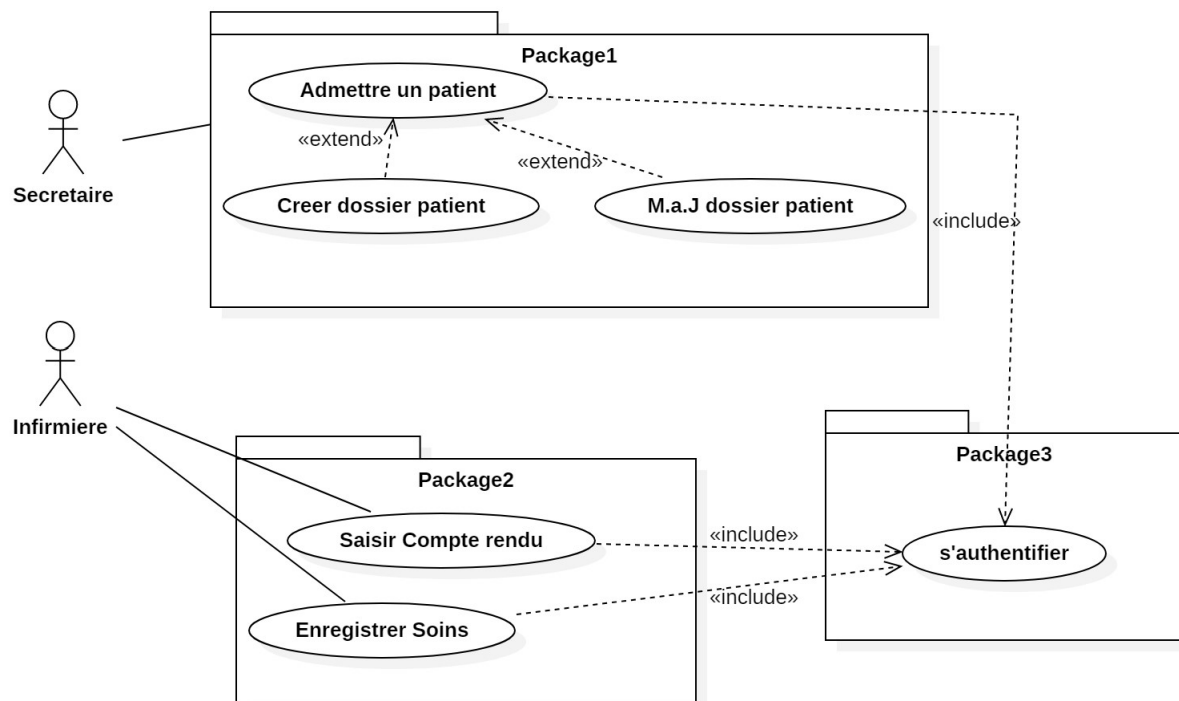


Figure 3: Identification des relations.

4.5 Description textuelle des Cas d'utilisation

Le diagramme de cas d'utilisation décrit les fonctionnalités du système à développer du point de vue des acteurs, mais ne décrit pas de façon détaillée l'interaction des acteurs avec les cas d'utilisation ou comment le cas sera réalisé. De ce fait, une description des cas d'utilisation est indispensable pour comprendre la chronologie d'actions exécutées par les acteurs et le système pour réaliser un cas donné. Il est recommandé de spécifier les cas par des fiches descriptives informelles qui servent de support de documentation et communication avec les utilisateurs. Généralement, la description textuelle comprend les éléments suivants:

- **Identification du cas** : indique le nom du cas et les acteurs impliqués (principaux et secondaires) dans sa réalisation, et résume son objectif.
- **Pré-conditions**: Il s'agit des conditions obligatoires pour le déclenchement du cas.
- **Post-conditions**: modifications apportées au système à la fin du cas.
- **Scénario nominal** : décrit le déroulement idéal des actions, où tout va pour le mieux sans échec.
- **Scénarios alternatifs**: sont des variantes du scénario normale.
- **Scénarios d'exception**: décrivent ce qui se passe lors d'une erreur.

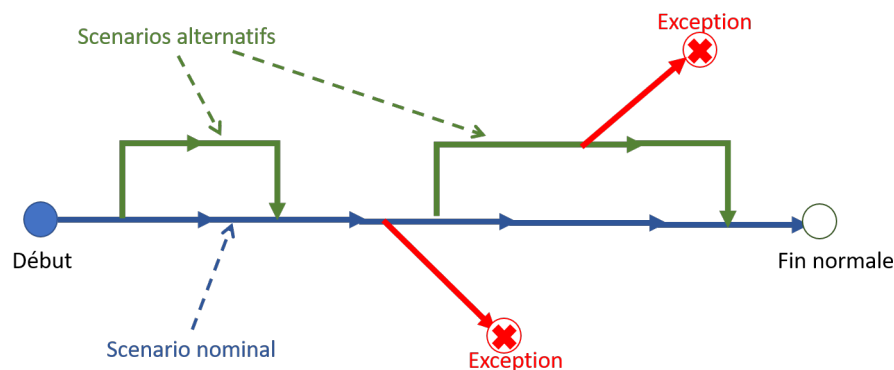


Figure 4: Scénarios d'un cas d'utilisation.

4.6 Exemple d'une fiche descriptive

Nom du cas	Admettre un patient
Type	Principal
Acteurs	Secrétaire
Objectif	Permet à la secrétaire d'attribuer une chambre à un patient.
Pré-condition	La secrétaire doit être authentifiée
Scénario nominal	<ol style="list-style-type: none"> 1) La secrétaire clique sur le bouton « Admettre un patient » 2) Le système lui affiche un formulaire contenant les renseignements sur l'hospitalisation du patient (dates entrée et sortie, service, etc.). 3) La secrétaire remplit le formulaire. 4) Le système valide le formulaire. 5) Le système affiche une liste des chambres libres dans ce service. 6) La secrétaire choisit une chambre. 7) Le système lui demande si elle veut attribuer la chambre a un patient existant ou de créer un nouveau patient. 8) La secrétaire choisit d'attribuer la chambre un patient existant. 9) Le système fait appel au cas "mettre à jour un patient". 10) Le système enregistre les informations de l'hospitalisation du patient.
Scénarios alternatifs	<p>A1. Le système ne trouve pas le patient. L'enchaînement démarre après le point 1 de la séquence nominale :</p> <ol style="list-style-type: none"> 2) Le système indique que le patient n'existe pas dans la BD. 3) Appel au cas "créer un dossier médical". 4) La séquence nominale reprend au point 8. <p>A2. Le système ne valide pas le formulaire. L'enchaînement démarre après le point 5 de la séquence nominale :</p> <ol style="list-style-type: none"> 6) Le système demande au secrétaire de remplir le formulaire. 7) La séquence nominale reprend au point 7.
Scénarios d'exception	<p>E1. Le système ne trouve pas une chambre libre. L'enchaînement démarre après le point 2 de la séquence nominale : Le système annule l'opération.</p>
Post-condition	Une chambre est attribuée au patient.

4.7 Description graphique des cas d'utilisation

La fiche descriptive d'un cas d'utilisation peut contenir plusieurs scénarios alternatifs et/ou d'exception. Il est alors difficile d'avoir une vision de l'ensemble des actions. Le diagramme diagrammes UML dynamiques donneront cette vision d'ensemble et apporteront un niveau supérieur de formalisation.

- **Diagramme de séquence système:** décrit le déroulement de la séquence d'actions des scénarios d'un cas d'utilisation en considérant le système comme une boîte noire. Le comportement du système est décrit du point de vue des acteurs, sans préjuger de comment il le réalisera.
- **Diagrammes d'activités :** permet de représenter les différents scénarios d'exécution d'un cas d'utilisation

5 Classement des cas d'utilisation et planification des itérations

Dans le cadre d'un développement incrémental et itératif, il est très utile de recourir au classement des cas d'utilisation pour définir les incréments et planifier les itérations. Cette classification (établie souvent par le chef de projet) prend lieu avant les activités de descriptions textuelle et graphique des cas d'utilisation en tenant compte des deux facteurs :

- la priorité fonctionnelle, déterminée par le client dans le but de livrer d'abord les cas d'utilisation les plus demandés.
- le risque technique, estimé par le chef de projet pour identifier les cas d'utilisation les plus critiques.

Il faut aussi prendre en considération d'analyser et développer en premier les cas primaires, principalement ceux mettant en cause l'acteur principal. Les cas secondaires et les cas optionnels seront déterminés ultérieurement. Cependant, ces critères pouvant être contradictoires:

- Si la priorité est haute et le risque également, il faut planifier le cas d'utilisation dans une des toutes premières itérations.
- Si la priorité est basse et le risque également, on peut reporter le cas d'utilisation à une des toutes dernières itérations.
- Les choses se compliquent lorsque les deux critères sont antagonistes ! Le chef de projet doit alors décider en pesant le pour et le contre.

Exemple

Cas d'utilisation	Priorité	Risque	Itération
Admettre un patient	Haute	Haut	1
Créer le dossier médical d'un patient	Haute	Moyen	2
Mettre à jour le dossier médical d'un patient	Moyenne	Bas	5
S'authentifier	Basse	Moyen	6
Saisir compte rendu d'une consultation d'un patient.	Haute	Bas	3
Enregistrer les soins offerts à un patient.	Haute	Bas	4

6 Production des maquettes IHM

Une maquette est un produit jetable donnant aux utilisateurs une vue concrète mais non définitive de la future interface de l'application. Il s'agit de simples dessins d'écrans et descriptions de contenu de fenêtres réalisées avec n'importe quel outil graphique. Les maquettes représentent un moyen de communication avec le client et les utilisateurs et aident à la description et la validation des cas d'utilisation. De ce fait, elles sont élaborées si tôt dans le processus de développement pour provoquer des retours de la part des utilisateurs.

Admettre un patient

Nom

Prenom

Date d'entree

Date de sortie

Service

Chambre disponible

Lits disponible

Admettre le patient

Valider

Figure 5: Exemple d'une maquette.

7 Conclusion

L'étude préliminaire et la spécification des besoins sont des activités très difficiles a cause de divers facteurs:

- **Problèmes de compréhension:** Les développeurs et le client na parle pas le même langage.
- **Problèmes de volatilité :** Une spécification valide peut ne plus l'être après une période de temps
- **Problèmes humains:** multiplicité des intervenants (client, utilisateurs, développeurs...), conflits, rétention d'information, etc.

Références

- GABAY, Joseph et GABAY, David. UML 2 Analyse et conception: Mise en œuvre guidée avec études de cas. Dunod, 2008.
- ARLOW, Jim et NEUSTADT, Ila. UML 2 and the unified process: practical object-oriented analysis and design. Pearson Education, 2005.
- Pascal Roques. Les cahiers du programmeur, UML2 Modéliser une application web. 4eme édition. EYROLLES, 2008.
- UP : Unified Process, ÉDITIONS EYROLLES
<https://sabricole.developpez.com/uml/tutoriel/unifiedProcess/LIV>, consulté le 29-11-2020.
- LETHBRIDGE, Timothy C. et LAGANIÈRE, Robert. Object-Oriented Software Engineering: Practical Software Development using UML and Java, 2004.