

## **Projet e-Crise**

"e-Crise" est un projet qui a pour but d'assister les interventions dans des situations de crise par l'informatique ou plus exactement des TIC (Technologies de l'Information et de la Communication). La notion de « crise » peut être plus ou moins clairement identifiée. Dans ce qui suit nous considérerons qu'il peut s'agir d'accident de la circulation plus ou moins important (carambolage dû au verglas sur l'autoroute impliquant de nombreux véhicules), déraillement d'un train, glissement de terrain, explosion accidentelle de gaz, pollution chimique accidentelle, .... Dans ces différents cas, de nombreux acteurs sont concernés (pompiers, secouristes, unités spécialisées en traitement de fuites de gaz ou chimique, ...). Dans tous les cas, les intervenants travaillent en équipes qui sont d'une part homogènes (même corps de métier : pompiers, secouristes, ...) et d'autre part hétérogènes (communication entre pompiers et secouristes, par exemple).

### **Organisation d'une gestion de crise, principaux aspects :**

- Processus : Analyser, organiser, traiter, évacuer, soigner, documenter
- Acteurs : chef, trieur, infirmier, médecin, transporteur, documentalistes
- Actions : identifier, quadriller le terrain, photographier, soigner, préparer et transporter, soigner, renseigner, documenter...
- Outils : diagnostiquer, todo list, bibliothèque de photos, base de données des personnes, formulaires de documentation

# Sommaire

## I/ Le cahier des charges

1. *Présentation générale*
2. *Définition des objectifs et des contraintes*
3. *Conditions de réalisation*
4. *Manuel d'utilisation*
5. *Manuel d'exploitation*
6. *Dossier de recettes*

## II/ Les spécifications :

1. *Texte de cadrage selon le plan type :*
2. *Modélisation UML*
  - a. *Diagrammes de cas*
  - b. *Diagrammes de séquence*
  - c. *Diagrammes de collaboration*
  - d. *Diagrammes d'état*
  - e. *Diagramme de classes*

## III/ Conception d'architecture :

1. *Diagramme d'architecture*
2. *Tests d'intégration*

## 1/ Le cahier des charges

### *1. Présentation générale*

- présentation de l'organisme et de ses activités

Nous sommes une entreprise qui fabrique des shampoings sur un site de production de surface 4500 m<sup>2</sup>. Chaque jour nous fabriquons plus de 10 000 shampoings par jour. Notre site comprend différents pôles de production correspondant aux différentes phases de fabrication du produit. La fabrication est quasiment automatisée grâce à des machines. Un opérateur est affilié sur chaque machine et se charge de la faire fonctionner, vérifier son bon fonctionnement et avertir d'une panne.

### *2. Définition des objectifs et des contraintes*

#### 2.1 Contexte

- problème dans son contexte habituel

Le site est un gros site de production qui doit faire face à différentes crises de taille différente. Les petites crises correspondent à la panne d'une machine sur le site de production qui peut parfois arrêter une chaîne complète pendant toute la durée de la réparation, à des accidents de travail (de différente gravité), à un manque d'approvisionnement en matières premières d'une chaîne de production ou encore un incident grave sur le site de production (incendie, grosse casse,...) Dans ce cas, les opérateurs ne savent souvent pas comment réagir, qui appeler pour résoudre le problème. Cela aboutit à un manque de coopération entre les différents acteurs et augmente fortement le temps de gestion d'une crise quelque soit son ampleur. Une entreprise de notre taille se doit d'optimiser la gestion de crise pour que son site de production soit le plus productif possible. Nous souhaitons alors mettre en place un service qui gère et coordonne les crises entre les opérateurs, le service de maintenance, les fournisseurs, les constructeurs et autres acteurs extérieurs.

- bilan de l'existant

Pour le moment, il n'existe pas de service de gestion de crise pour coordonner les personnes lors d'une crise.

Si une machine tombe en panne, l'opérateur ne sait pas comment réagir. Elle appelle le service maintenance (qui comprend des automaticiens, électriciens, mécaniciens, plombiers, ...). Seulement, il ne sait pas à qui s'adresser pour qu'on le dépanne. De plus, il est difficile pour lui de tomber sur une personne qui connaisse la machine, qui soit disponible,... Cela augmente le temps de réparation. De plus, il n'existe pas d'historique de panne qui pourrait l'aider lui ou la personne de maintenance à résoudre la panne plus facilement.

Si un incident se passe sur le site, les différents acteurs ne sont pas coordonnés pour régler le problème (ex : pompiers, service constructeur, ...). Il n'y a pas de protocole prévu pour gérer la crise.

- aspects qualitatif et quantitatif
  - (taille du problème, fréquence d'utilisation,...)

Ceci est un problème important. La gestion de crise doit être opérationnelle 24h. Elle sera utilisée quasiment en permanence puisque les petites pannes sont régulières étant donné le nombre important de machines.

#### 2.2 Définition fonctionnelle (Attention tous les points de vue ne sont pas tous toujours pertinents, notamment le premier)

- Analyse mathématique du problème

#### Gestion de crise

- description du phénomène étudié
- modèle mathématique
- modèle numérique
- Analyse basée sur les principaux traitements à assurer
  - traitements et leurs entrées-sorties
  - conditions d'utilisation

Le système de gestion de crise doit comprendre un trieur que l'opérateur (ou du moins la personne qui se rend compte de la crise) appelle au démarrage d'une crise. Ensuite, soit le trieur le met en relation avec les personnes adéquates (ex: avec le service de maintenance spécialisé dans le domaine de la panne (électricité, automatisme,...)), soit il coordonne les différentes personnes intervenant dans la résolution du problème.

- erreurs prévisibles
- Le trieur n'appelle pas les bonnes personnes pour régler le problème (ex : il se trompe de service (électricité, automatisme, ...) pour réparer la machine en panne.
- La crise est exceptionnelle : elle n'a pas été prévue et le trieur ne sait pas quel protocole appliquer.

- Analyse par inventaire des données manipulées
- nature et caractéristiques

Les données manipulées sont principalement les numéros des acteurs utiles à la résolution du problème (service maintenance notamment ou acteurs extérieurs). Il doit également y avoir une base de données qui comprend les notices / manuels d'utilisation, un historique de pannes (compte rendu de panne), numéro du constructeur, numéro de la personne de maintenance spécialiste de cette machine si elle existe, pour chaque machine.

A chaque crise, un compte rendu doit être créé pour l'insérer dans la base de données. Ainsi, on a un historique des crises qui permet de gagner du temps la fois suivante.

- volume et fréquence d'utilisation

Il y a une documentation pour chaque machine. Elles sont utilisées assez régulièrement puisque des petites pannes ont lieu chaque semaine sur chaque type de machine.

- échanges avec le monde extérieur

Cette base de données n'est pas accessible par le monde extérieur. Seul le personnel de l'usine a accès à cette base de données (trieur, opérateurs, personnels de maintenance, infirmières, chefs d'équipe,...) et chaque personne a accès à sa to-do list.

#### 2.3 Définition de l'environnement

- interface homme-machine (dispositifs, ergonomie,...)

Le logiciel doit être facile d'utilisation (utilisable par des personnes non nécessairement qualifiées en matière d'informatique).

- interface entre logiciels (couplage SGBD,...)

## Gestion de crise

Le logiciel doit contenir une base de données qui comprend notamment des documents sur les différentes machines de l'entreprise (manuel d'utilisation et historique de pannes).

- mode de fonctionnement (interactif, traitement par lots à distance,...)

Le logiciel doit être interactif. L'utilisateur doit pouvoir communiquer et donner des informations au logiciel. Le logiciel doit être en mesure de réagir et se comporter en fonction des informations qu'il a reçues de l'utilisateur.

- performances (temps de réponse, taille, précision,...)

Le logiciel doit avoir un temps de réponse rapide. Il doit réagir immédiatement aux différentes requêtes de l'utilisateur.

### 2.4 Définition de la procédure de recette

- conditions de recette (date, lieu, environnement, ...)

La recette doit avoir lieu sur le site de production de l'entreprise le 30 janvier 2010. Les développeurs doivent expliquer comment fonctionne le logiciel au client et fournir les documents.

- tests fonctionnels

Les développeurs doivent effectuer des tests pour des pannes sur tout type de machine avec différents cas de résolution (mise en correspondance avec une personne de maintenance adéquate et réparation par téléphone, sur place ou appel d'une personne du constructeur SAV) mettant en œuvre différents services de la maintenance ainsi que pour la gestion de tout autre type de crises recensées par le client.

- documents à fournir

Les développeurs doivent fournir un document qui expliquent la démarche de développement du logiciel ainsi qu'un manuel d'utilisation, un manuel d'exploitation et un manuel de maintenance qui comprend les différents problèmes d'utilisation et erreurs (avec leur protocole de résolution).

## **3. Conditions de réalisation**

### 3.1 Engagement du maître d'ouvrage (client)

- responsabilité

Il s'engage à acheter le logiciel à la fin de sa réalisation et à ne pas modifier sa demande et le cahier des charges en cours de réalisation du projet. Celle-ci est fixée et ne peut pas évoluer.

- moyens (données, informations, connaissances,...)

Le client doit fournir un exemple type de notice d'une machine pour que celle-ci soit traitée dans le logiciel, l'organigramme de l'entreprise (différentes personnes qui seront actives dans la gestion de crise).

### 3.2 Engagement du maître d'œuvre (développeur)

- contraintes physiques (matériel de développement, ...)

Aucune contrainte physique

- contraintes de réalisation (langage, outils, méthodes, atelier, normes,...)

La base de données doit être rédigée sous PhpMySQL et la todolist relative à la gestion de crise en php / html. Le logiciel doit être développé grâce à Wampserver.

- fournitures (logiciels, documentations, formation,..)

Les développeurs doivent fournir le manuel d'utilisation, manuel d'exploitation et manuel de maintenance du logiciel et former une personne à l'utilisation du logiciel ainsi qu'à sa maintenance.

- services supplémentaires

Les développeurs doivent rester disponibles pendant une durée minimale de deux ans afin de répondre aux questions des utilisateurs et maintenir le logiciel en cas de difficultés de la personne de l'entreprise en charge du logiciel. De plus, les développeurs doivent pouvoir répondre à la demande d'un client en vue d'une amélioration du logiciel.

### 3.3 Suivi de la réalisation

- conditions

Les développeurs doivent montrer régulièrement leur avancement pour s'assurer de la compatibilité avec la demande du client.

- réunions

Toutes les semaines est organisée une réunion entre le client et le développeur pour suivre l'avancement du problème et s'assurer que le travail effectué correspond à la demande du client.

### 3.4 Conditions de réception

- Procédures de réception

Les développeurs se rendent sur le site du client et mettent en place le logiciel de gestion de crise afin de vérifier son bon fonctionnement et donner les informations utiles et documents à l'administrateur en charge du logiciel dans l'entreprise. Les développeurs doivent effectuer des tests correspondant aux différents cas d'utilisation pour vérifier le bon fonctionnement du logiciel.

- Procès verbal de réception

Le client doit remplir un papier qui justifie la livraison du logiciel et la conformité avec le cahier des charges.

- Contexte administratif et juridique

Le logiciel appartient à l'entreprise qui en a fait l'acquisition.

Gestion de crise

○ <b>Facteurs d'exploitation</b>	Sans intérêt	Peu important	Moyennement important	Très important
○ <b>Confidentialité</b> : aptitude d'un logiciel à être protégé contre tout accès par des personnes non autorisées.			Les données ne doivent être accessibles que par le personnel	
• <b>Couplabilité</b> : aptitude d'un logiciel à être couplé à un autre logiciel.	Cela n'est pas nécessaire			
• <b>Efficacité</b> : aptitude d'un logiciel à minimiser l'utilisation des ressources disponibles.		Il doit pouvoir avoir accès à la base de données		
• <b>Maniabilité</b> : aptitude d'un logiciel à être convivial et facile d'emploi pour l'utilisateur auquel il est destiné.				Il doit être facile d'utilisation pour ne pas perdre de temps lors de la recherche de solutions
○ <b>Robustesse</b> : aptitude d'un logiciel à conserver un comportement conforme aux besoins exprimés en présence d'événements non souhaités ou non prévus		Le logiciel doit prévoir la possibilité de remplir des champs par des items non proposés (une case « autre » doit être prévue pour les cas non répertoriés)		

Gestion de crise

○ <b>Facteurs de maintenance</b>	Sans intérêt	Peu important	Moyennement important	Très important
○ <b>Maintenabilité</b> : aptitude d'un logiciel à faciliter la localisation et correction d'erreurs résiduelles.				L'administrateur doit être facilement en mesure de corriger les erreurs résiduelles
○ <b>Adaptabilité</b> : aptitude d'un logiciel à faciliter la suppression ou l'évolution de fonctionnalités existantes ou l'adjonction de nouvelles fonctionnalités.				L'administrateur doit être en mesure de faire évoluer les bases de données facilement
○ <b>Portabilité</b> : aptitude d'un logiciel à minimiser les répercussions d'un changement d'environnement logiciel et matériel.			Le logiciel doit être utilisable avec n'importe quel type de navigateur web	
○ <b>Testabilité</b> : aptitude d'un logiciel à permettre la vérification de son fonctionnement.		Cela est peu important		
○ <b>Observabilité</b> : aptitude d'un logiciel à fournir des informations sur son fonctionnement.			Le logiciel doit fournir les informations sur son mode de fonctionnement (notamment lors de l'identification des crises et du remplissage du compte-rendu	
○ <b>Réutilisabilité</b> : aptitude d'un logiciel à être intégré en partie ou en entier dans d'autres logiciels.	Cela n'est pas demandé			

#### 4. Manuel d'utilisation

Le programme « Gestion de crise » devra aider le personnel de l'entreprise à gérer tout type de crise qu'elle qu'en soit la gravité, l'importance, les acteurs ou matériels mis en jeu. Il proposera des protocoles de gestion efficaces permettant de minimiser le temps de résolution.

Le logiciel indiquera aux personnes intervenant successivement dans la résolution de la crise le protocole à suivre : le logiciel recense les différents types de crises qui peuvent arriver sur le site de l'entreprise :

- accident corporel
- incident sur site (incendie, inondation, ...)
- rupture d'approvisionnement en matière première
- panne d'une machine de production
- ...

Lors du déclenchement d'une crise, la personne qui se rendra compte de la crise devra immédiatement prévenir le trieur.

→ Le trieur devra poser une certaine quantité de questions à l'opérateur qui l'a contacté pour identifier le type de crise. Pour l'aider, une base de données interactive propose au trieur un compte rendu de crise qu'il doit commencer à remplir (le compte rendu est créé pas à pas comme décrit ci-après). Le logiciel propose une série de questions avec un champ à remplir successivement où le trieur doit choisir un item parmi ceux proposés par le logiciel. A chaque itération, le logiciel propose une nouvelle question avec une liste de choix de plus en plus précis (en fonction des réponses précédentes) jusqu'à d'identifier précisément le type de crise. Une fois la crise bien identifiée, il propose un protocole adéquat. Prenons deux exemples :

##### Exemple 1

Un opérateur appelle le trieur pour l'informer d'un accident corporel

Le compte rendu à remplir par le trieur est le suivant :

Type de crise : accident corporel

Nombre de personnes blessées : 10

Gravité des blessures : très importante

Le logiciel lui donne la marche à suivre :

- Trier  
  - Prévenir l'infirmier
  - Transmettre le compte rendu
- Infirmière  
  - Appeler le SAMU
  - Récolter les informations utiles sur les personnes blessées
  - Compléter et archiver le compte rendu après l'intervention du SAMU

##### Exemple 2

Un opérateur appelle le trieur pour l'informer d'une panne sur sa machine.

L'ébauche du compte rendu est la suivante :

Type de crise : panne d'une machine de production

Numéro du poste de la machine : 5

Opérateur : Monsieur Raphaël Peigné

Nom de la machine : Automatica2000

Référence : A2000-5-12-9675

Type de panne : panne électrique

Priorité : 2 (faible)

Le logiciel fournit ensuite la marche à suivre pour chaque personne intervenant dans la résolution de la crise.

- Trieur
  - Prévenir la chef d'équipe concerné : il lui fournit les coordonnées (nom et numéro) du chef d'équipe de maintenance correspondant au secteur de la panne (secteur électricité)
  - Transmettre le compte-rendu
- Chef d'équipe
  - Affecter la tâche à un électricien de son équipe
  - Compléter et transmettre le compte-rendu au technicien en question
- Électricien
  - Récupérer le manuel d'utilisation et l'historique des pannes de la machine en question
  - Effectuer une réparation à distance en donnant les instructions à l'opérateur (par téléphone)
  - Compléter le compte rendu
  - Archiver le compte rendu
  - Prévenir son chef de la fin d'affectation à cette résolution de crise
- Chef d'équipe
  - Désaffecter l'électricien à cette tâche

Ensuite, le logiciel comprend une Todolist qui sera fait selon le modèle suivant. Chaque personne possède une todolist avec la liste des tâches qu'elle doit effectuer (appellation de la tâche, priorité, date de réalisation, durée, état (non commencée, en cours, finie)).

Il y a différentes catégories de personnes :

- les opérateurs : ils ont seulement accès à leur todolist, ils peuvent ajouter une tâche mais ne peuvent pas modifier celles qu'ils ont. Lorsqu'ils ont réalisé une tâche, soit ils préviennent leur chef d'équipe qui supprime la tâche après validation de sa réalisation, soit ils la suppriment eux-mêmes
- les chefs d'équipe : ils ont accès à leur todolist où ils peuvent ajouter, modifier, supprimer une tâche ainsi qu'à la todolist de toutes les personnes de leur équipe pour laquelle il peut également modifier la todolist (ajout, modification, suppression d'une tâche)
- l'administrateur qui gèrera le logiciel.

Le logiciel crée un diagramme de Gantt pour chaque todolist afin de mieux visualiser l'emploi du temps de la personne concernée.

Le logiciel crée un tableau des disponibilités pour chaque personne d'une équipe, accessible à toutes les personnes de l'équipe concernée, pour faciliter l'affectation des tâches et l'organisation de l'équipe.

Le logiciel devra également comporter une base de données qui comprend deux types de données :

- Un manuel d'utilisation de chaque machine comprenant également les informations du fournisseur. Celui-ci doit être accessible par tous les techniciens qui s'occupent de la maintenance des machines ainsi que par les opérateurs qui travaillent sur cette machine.

#### Gestion de crise

- Un historique des comptes rendus des crises de même type (Par exemple, pour une machine, on va avoir toutes les pannes l'ayant affectée avec un descriptif de la résolution de la panne)

→ Ces données seront accessibles par toutes les personnes de l'entreprise. Les utilisateurs pourront seulement ajouter des données. Pour modifier ou supprimer des documents, ils doivent faire appel à l'administrateur.

Pour pouvoir utiliser ce logiciel, chaque personne aura un identifiant et un mot de passe qu'elle devra rentrer afin de se connecter à la session qui la concerne.

## ***5. Manuel d'exploitation***

### Gestion du logiciel

Un informaticien est chargé d'assurer le bon fonctionnement du logiciel dans l'entreprise. Il doit être capable de gérer les erreurs d'utilisation, guider les utilisateurs en cas de problèmes. Il est en mesure d'apporter quelques modifications dans la base de données du logiciel : ajout/modification/suppression d'une personne, de documents dans la base de données,...

### Amélioration

Le client peut faire appel aux développeurs pour apporter des améliorations au logiciel. Pour cela, il devra exprimer son besoin sous forme d'un cahier des charges auprès de l'équipe de développement qui proposera alors une nouvelle version améliorée du logiciel.

**6. Dossier de recettes**

Date de la recette :

Lieu :

Personnes de l'entreprise cliente présentes :

- 
- 
- 

Personnes de l'entreprise de développement présentes :

- 
- 
- 

Durée de la recette :

Tests effectués :

- Cas d'utilisation n°1

Commentaires sur la conformité aux cahiers des charges

- Cas d'utilisation n°2

Commentaires sur la conformité aux cahiers des charges

- Cas d'utilisation n°3

Commentaires sur la conformité aux cahiers des charges

- Cas d'utilisation n°4

Commentaires sur la conformité aux cahiers des charges

- Cas d'utilisation n°5

Commentaires sur la conformité aux cahiers des charges

- Cas d'utilisation n°6

Commentaires sur la conformité aux cahiers des charges

- Cas d'utilisation n°7

Commentaires sur la conformité aux cahiers des charges

- Cas d'utilisation n°8

Commentaires sur la conformité aux cahiers des charges

Conclusion sur les tests effectués

## II/ Les spécifications :

### *1. Texte de cadrage selon le plan type :*

#### Description succincte du logiciel à concevoir

Le client demande un logiciel qui permet de gérer les crises de tout type (rupture de stock de matière première, panne sur une machine, accident corporel, incident structurel,...).

Le logiciel doit comprendre :

- une base de données à deux tables (manuels d'utilisation des machines avec les coordonnées des fournisseurs et historique des pannes)
- une base de données qui contient tout le personnel de l'usine avec des identifiants et mots de passe pour accéder à leur section du logiciel
- chaque personne accède à sa todolist et à son planning de travail. Suivant son statut, elle peut modifier, ajouter ou supprimer des tâches et avoir accès aux todolists des autres employés.
- des compte rendus types à remplir au fur et à mesure par le trieur, opérateur ou technicien de maintenance → celui qui remplit le compte-rendu doit répondre à une question en choisissant un item parmi plusieurs choix, la question suivante et les choix possibles dépendent des réponses précédentes. Le logiciel doit donc avoir des arbres descendants qui spécialisent les crises. En répondant aux questions, le trieur identifie précisément la crise à gérer. Le logiciel lui fournit alors la procédure à suivre pour gérer la crise (personnes à prévenir, personnes à contacter, pièces à commander, ...)

Lors d'une crise, la personne qui s'en rend compte informe le trieur qui se sert du compte rendu type décrit précédemment pour identifier la crise et lance ensuite le protocole de gestion de la crise fourni par le logiciel.

#### Critères de qualité

- **Banalité d'emploi** : le logiciel doit être utilisable grâce à n'importe quel navigateur internet
- **Communicabilité** : le logiciel doit fournir une interface facile d'utilisation d'accès pour l'utilisateur
- **Efficacité mémoire** : les bases de données doivent prendre le moins de place mémoire possible et les chemins d'accès doivent être simples
- **Efficacité temps d'exécution** : le temps de réponse à une requête doit être quasi immédiat
- **Expansibilité** : il doit être possible d'ajouter du personnel, de rajouter de la documentation sur les machines
- **Exploitabilité**: l'utilisateur doit pouvoir accéder facilement à sa session et utiliser les commandes qui lui sont possibles (ajout / modification / suppression de tâche) de façon simple.
- **Facilité d'apprentissage** : les utilisateurs doivent pouvoir savoir utiliser par eux-mêmes sans difficulté après une petite formation

## Gestion de crise

→ **Historique des accès** : l'administrateur doit avoir accès aux historiques des opérations effectuées. Chaque utilisateur doit pouvoir voir les opérations effectuées dans les 15 derniers précédents.

→ **Lisibilité** : le code doit être commenté et être organisé suivant l'architecture MVC. Les données doivent être lisibles (todolist facilement compréhensible, emploi du temps facilement compréhensible)

→ **Précision** : le logiciel doit bien identifier le type de crise suivant le compte rendu rempli par le trieur et fournir le bon protocole de gestion de crise

→ **Protection du code et des données** (en exploitation): il y a différents types de personnes qui ont des accès limités aux données

- l'administrateur a accès à toutes les données et toutes les sessions de tous les utilisateurs
- les chefs d'équipe ont accès aux todolists des personnes de leur équipe et peuvent ajouter / modifier / supprimer des tâches
- les techniciens de maintenance n'ont accès qu'à leur todolist et ne peuvent juste en ajouter
- seul le personnel de l'usine a accès aux bases de données (manuel d'utilisation et historiques de pannes)

→ **Protection du code et des données** (hors exploitation) : les utilisateurs doivent entrer leur identifiant et mot de passe pour pouvoir avoir accès au logiciel. Toute personne en dehors de l'entreprise n'a aucun accès

→ **Simplicité** : l'interface ne doit que contenir des données utiles sans ajout superflu

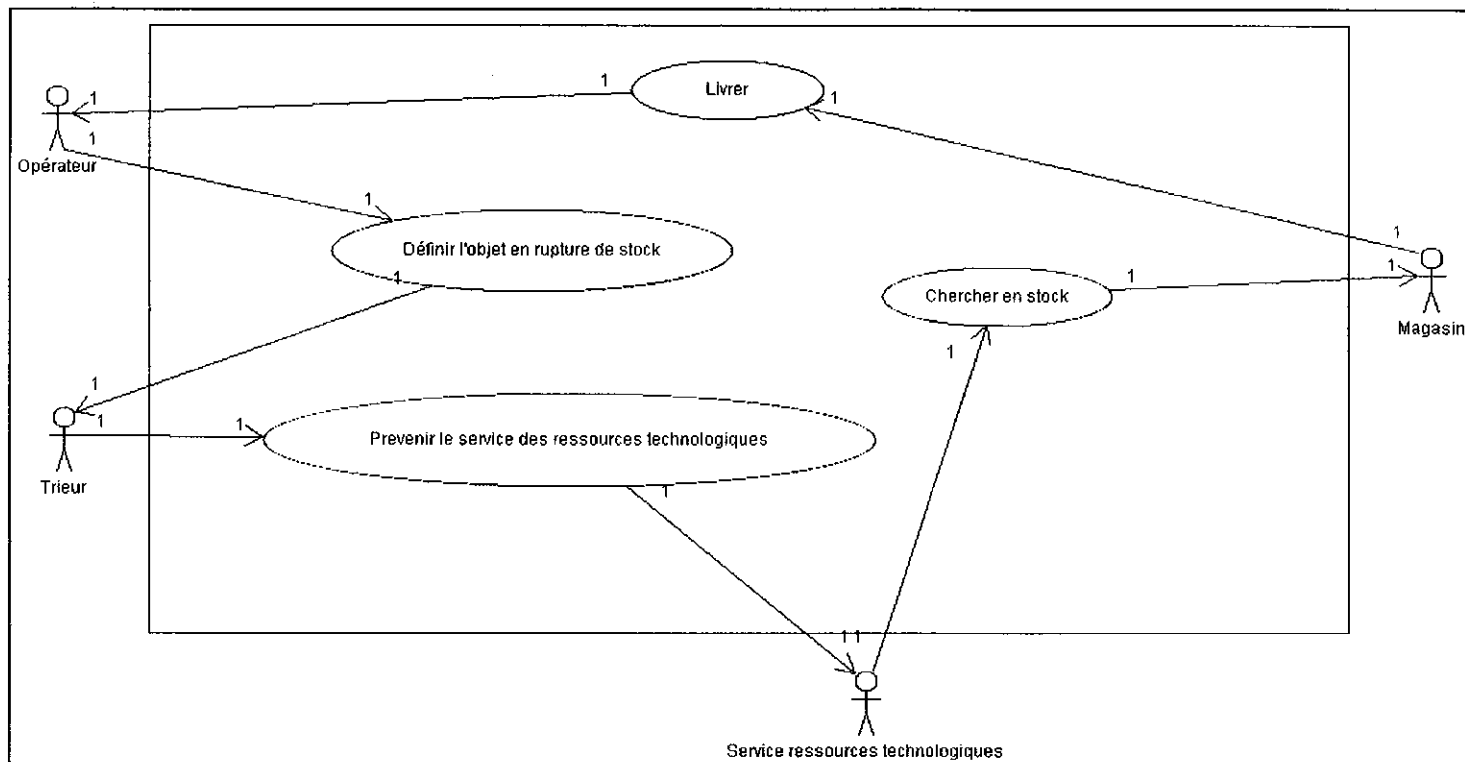
→ **Tolérance aux fautes** : lorsque le trieur remplit le compte rendu au début de la crise, il doit pouvoir remplir un champ « autre » si la crise n'était pas prise en compte (si une crise non prise en compte par le logiciel arrive, elle doit être rentrée dans le logiciel avec un protocole une fois gérée)

## 2. Modélisation UML :

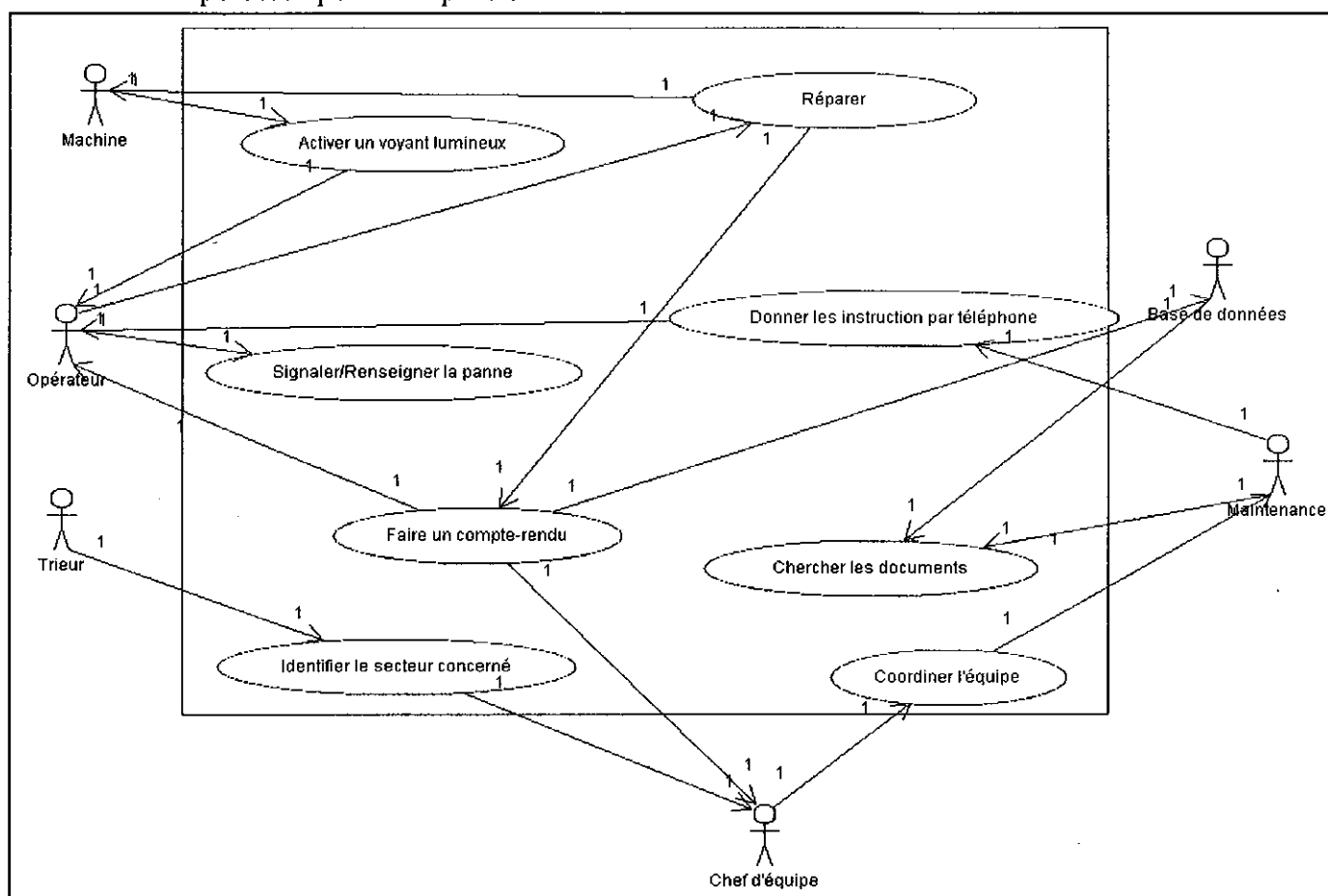
### a. Diagrammes de cas :

## Gestion de crise

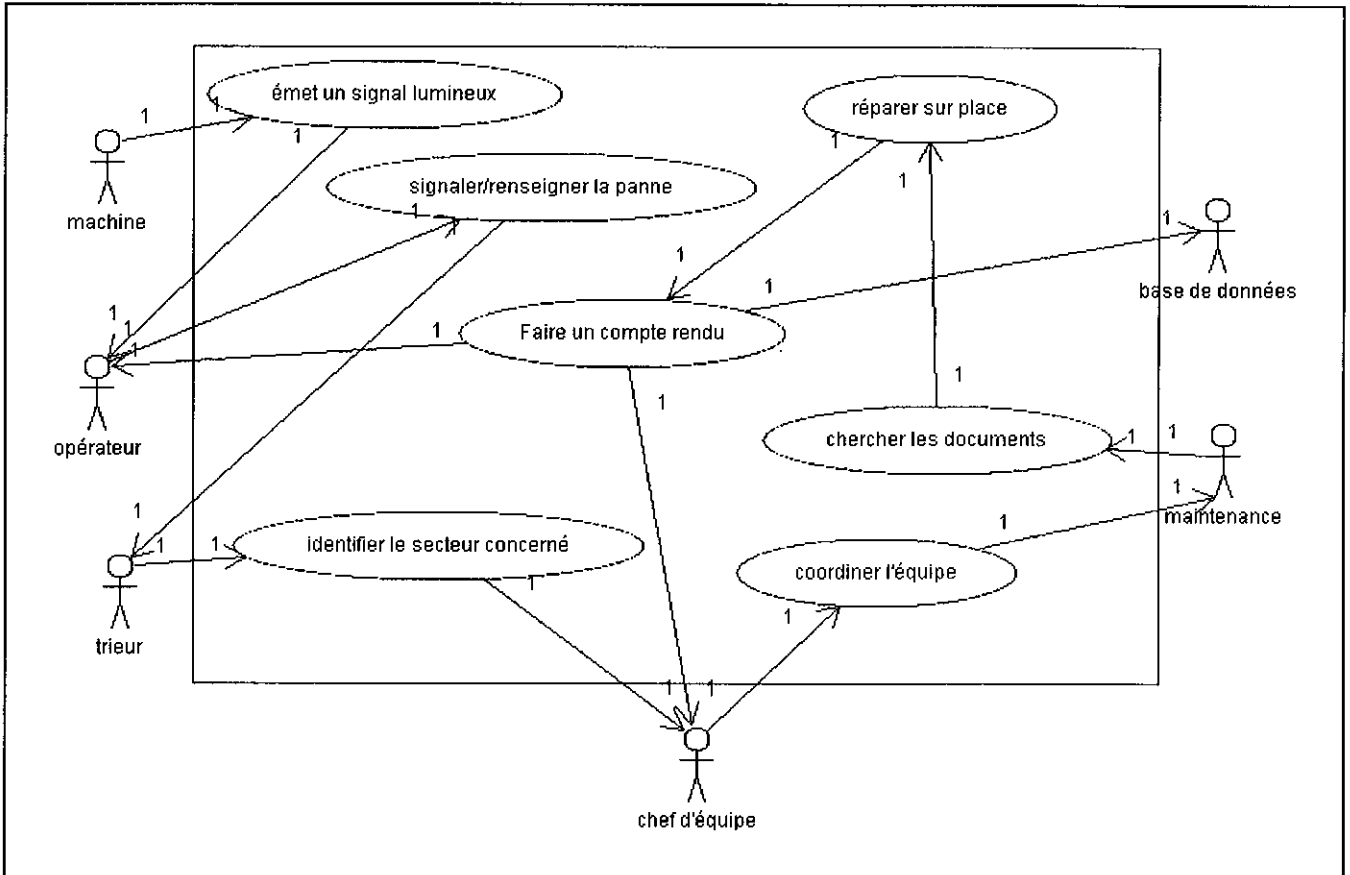
1) Il manque des pièces pour la production et il faut les demander au magasinier.



2) La machine est en panne et la maintenance donne les instructions par téléphone à l'opérateur pour la réparer.



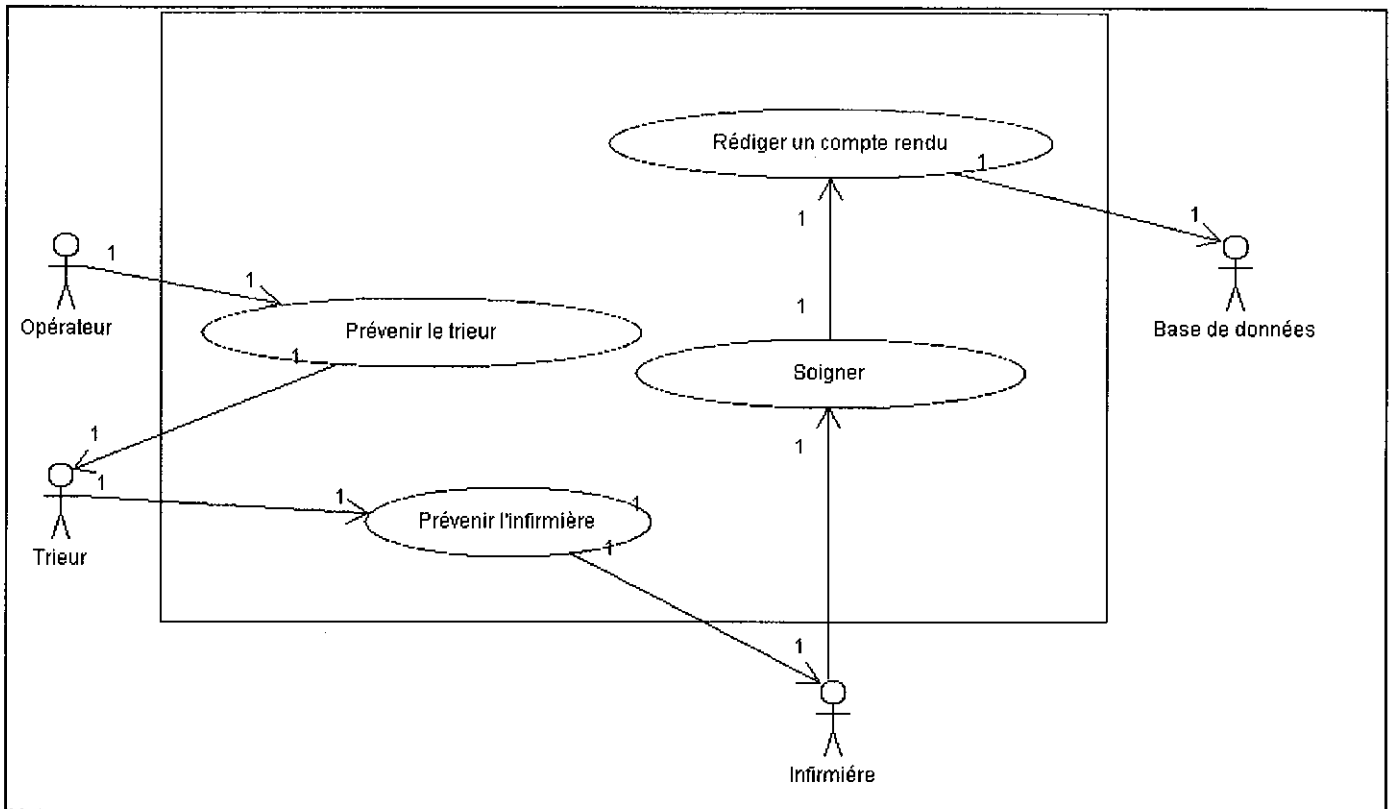
3) La machine est en panne et la maintenance vient réparer la machine sur place.



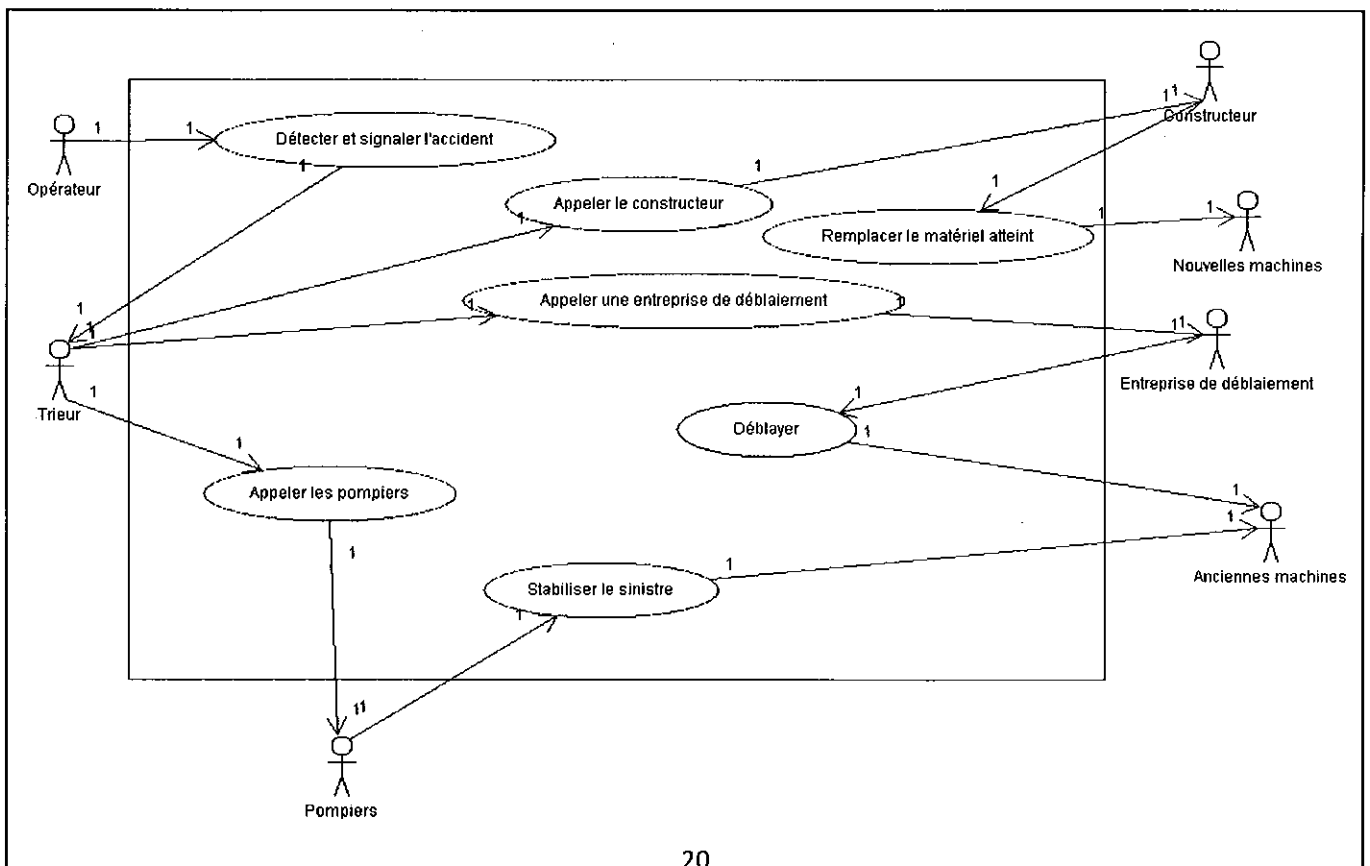


## Gestion de crise

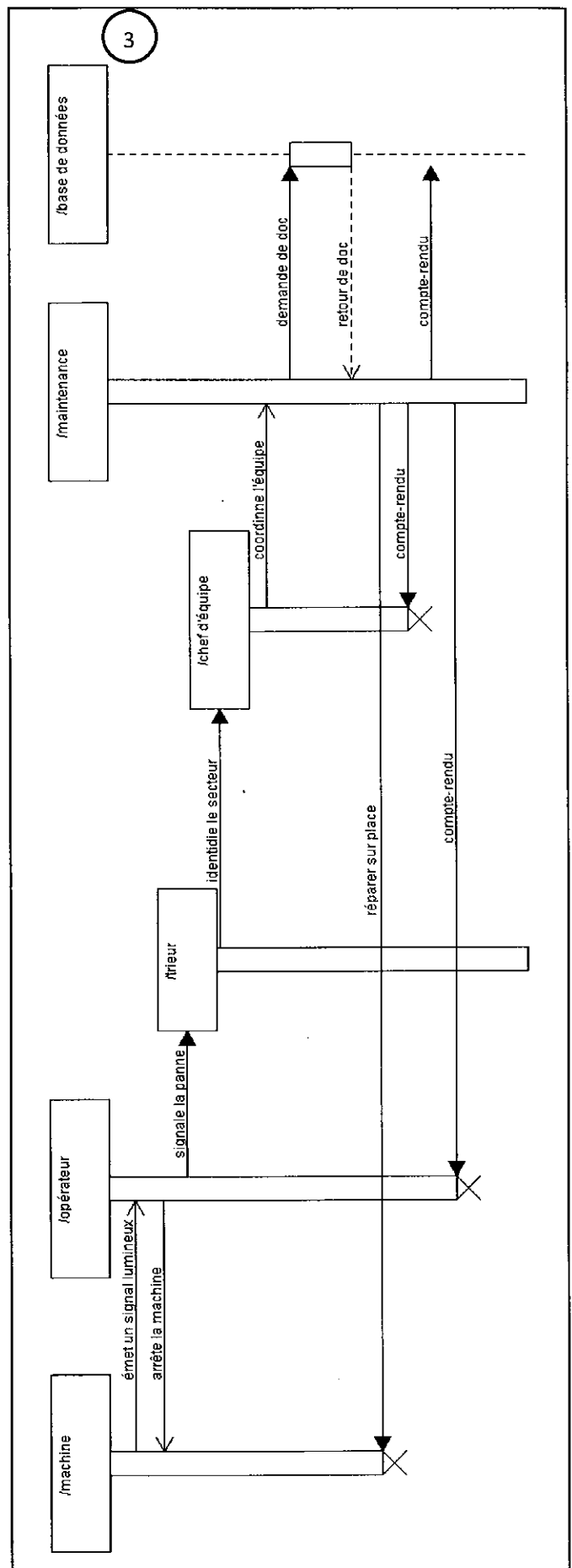
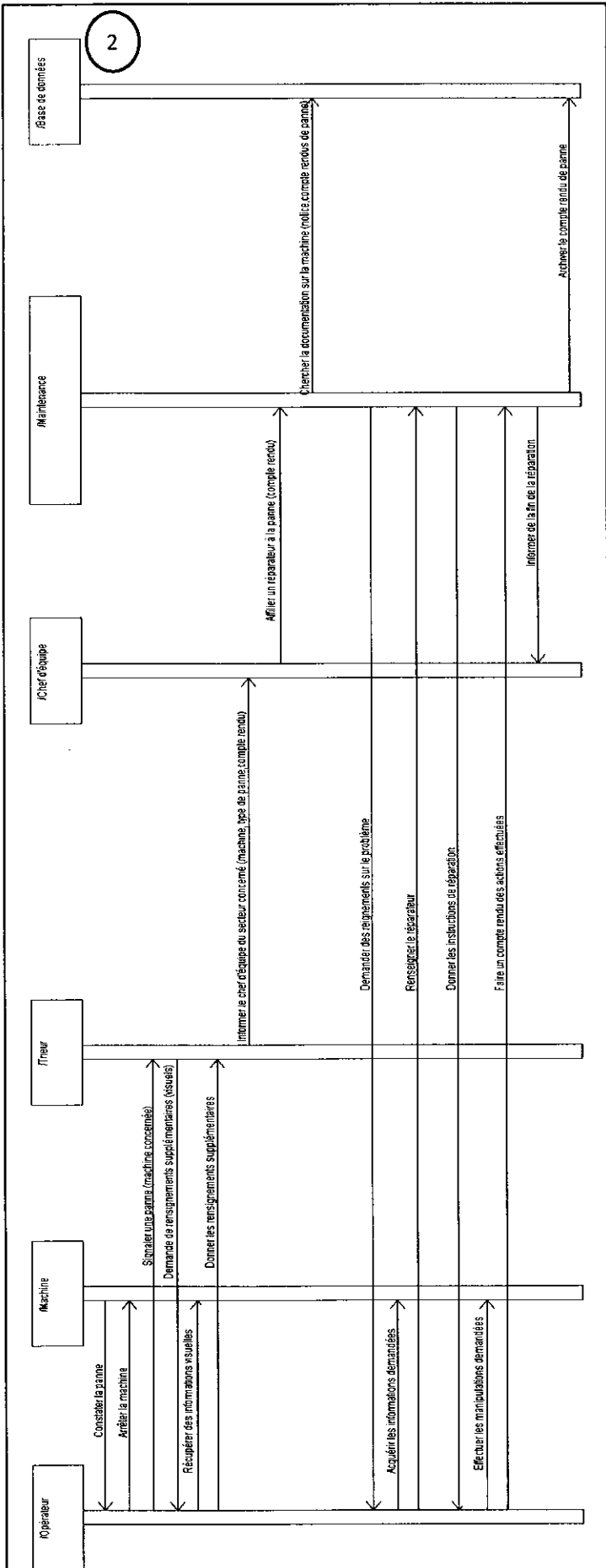
### 6) L'opérateur se blesse et l'infirmière intervient.

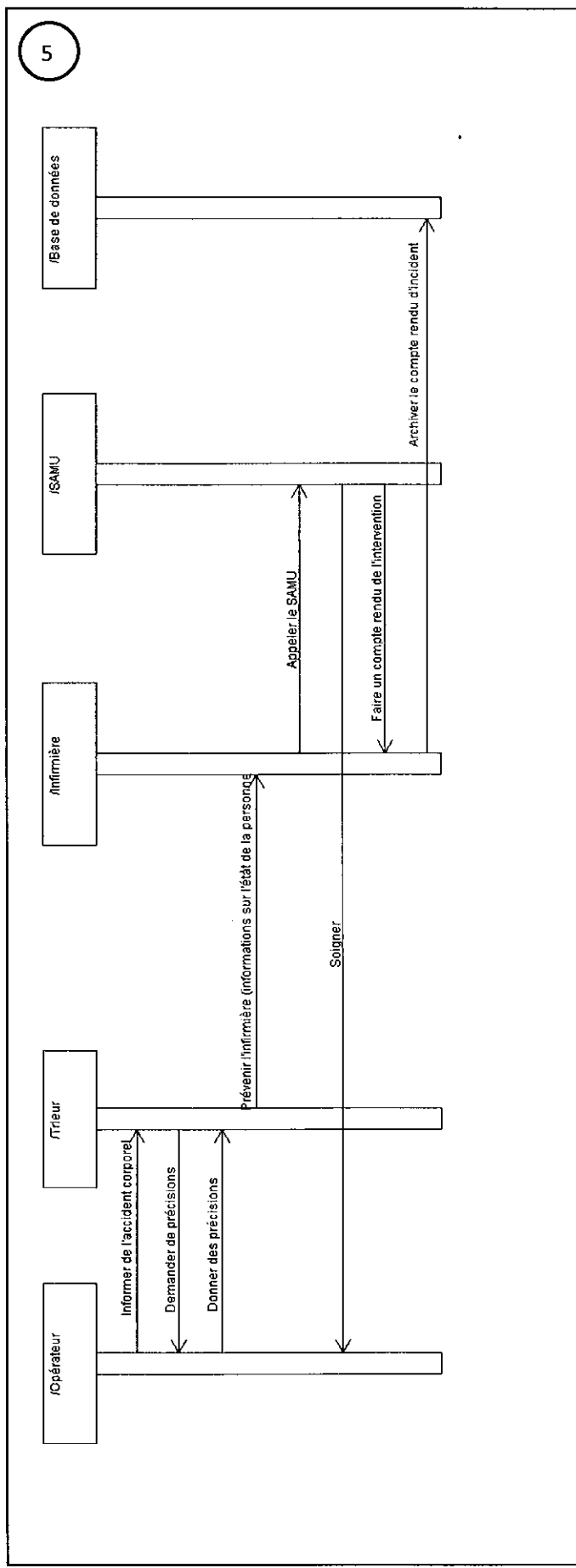
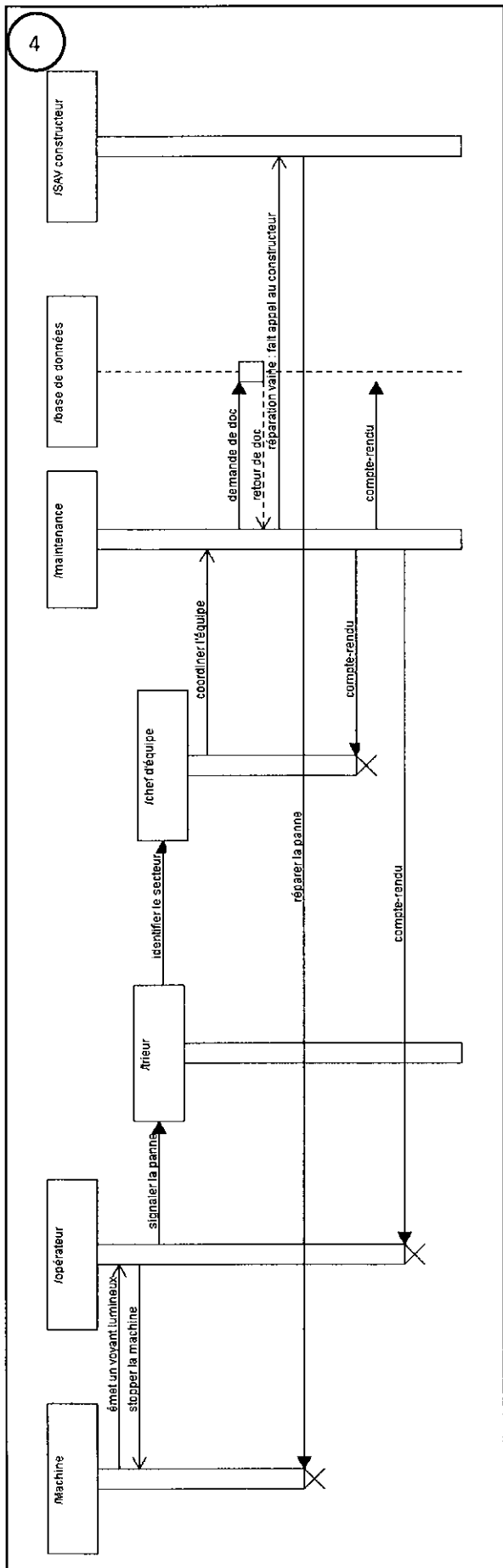


### 7) Un accident survient et les pompiers interviennent.

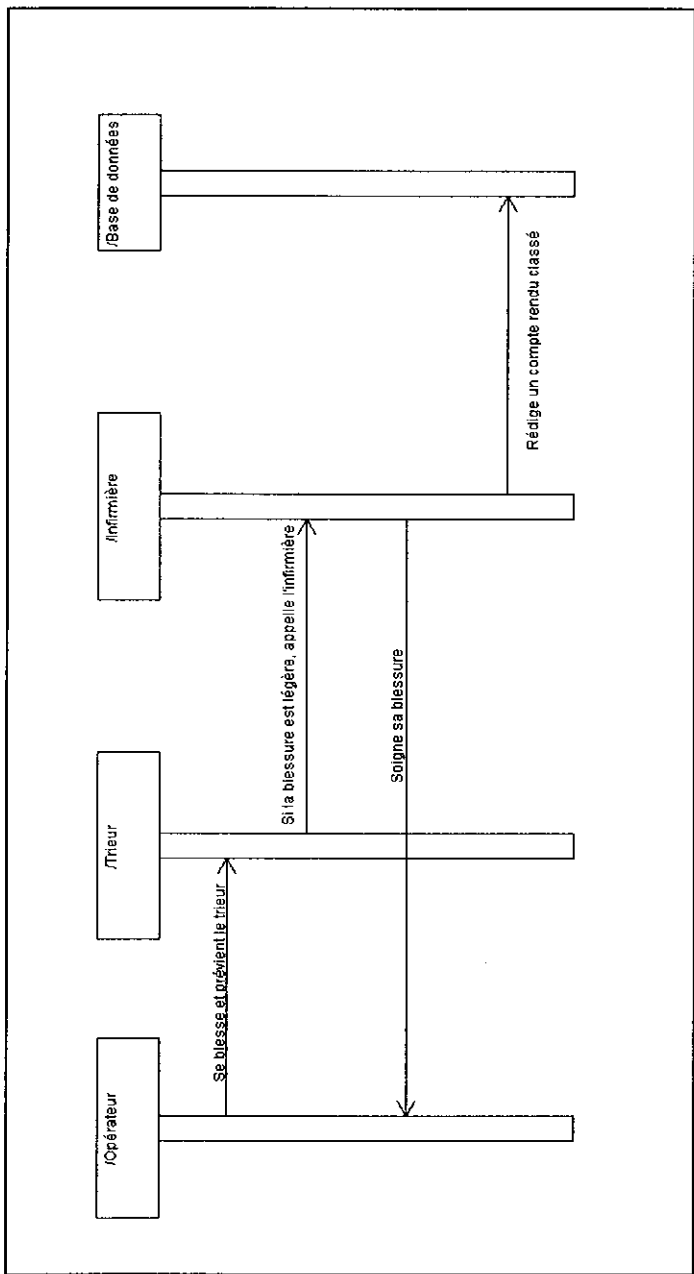




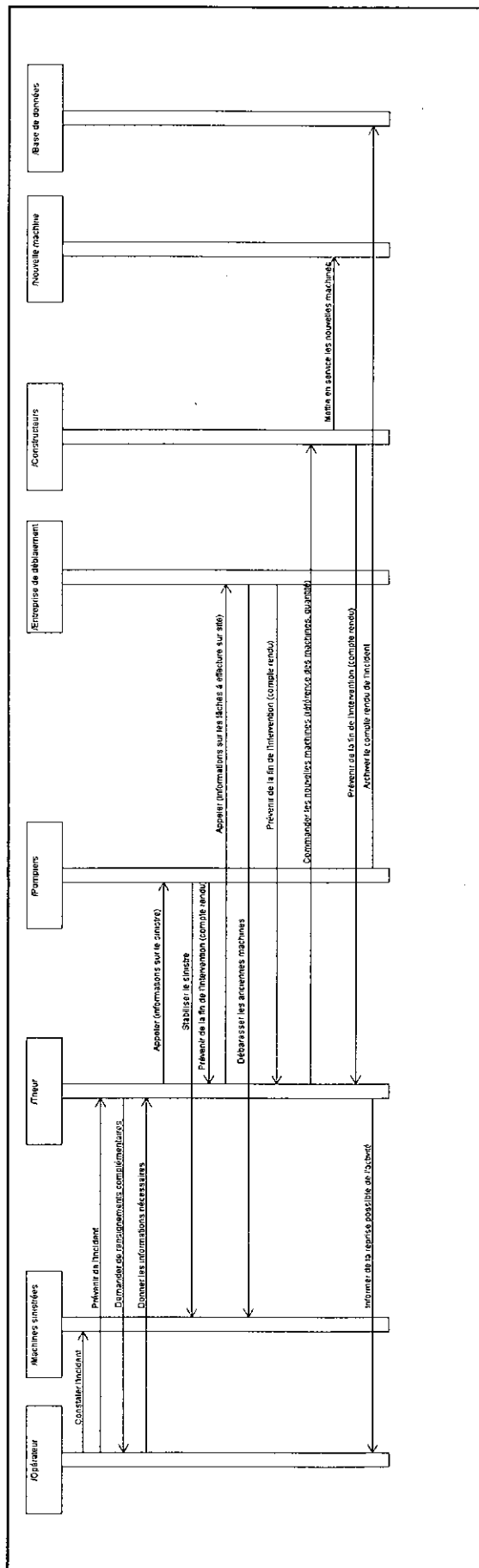


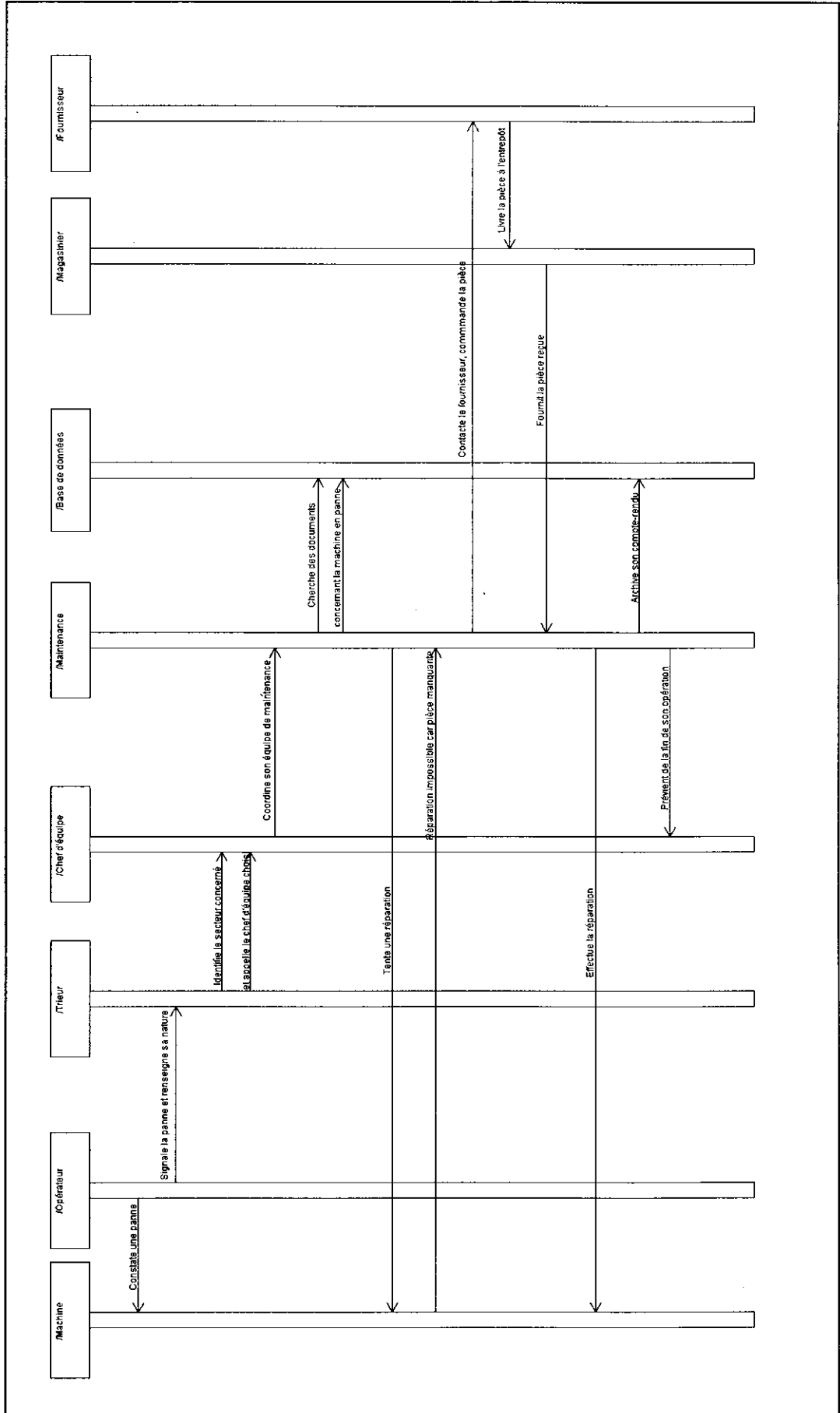


6

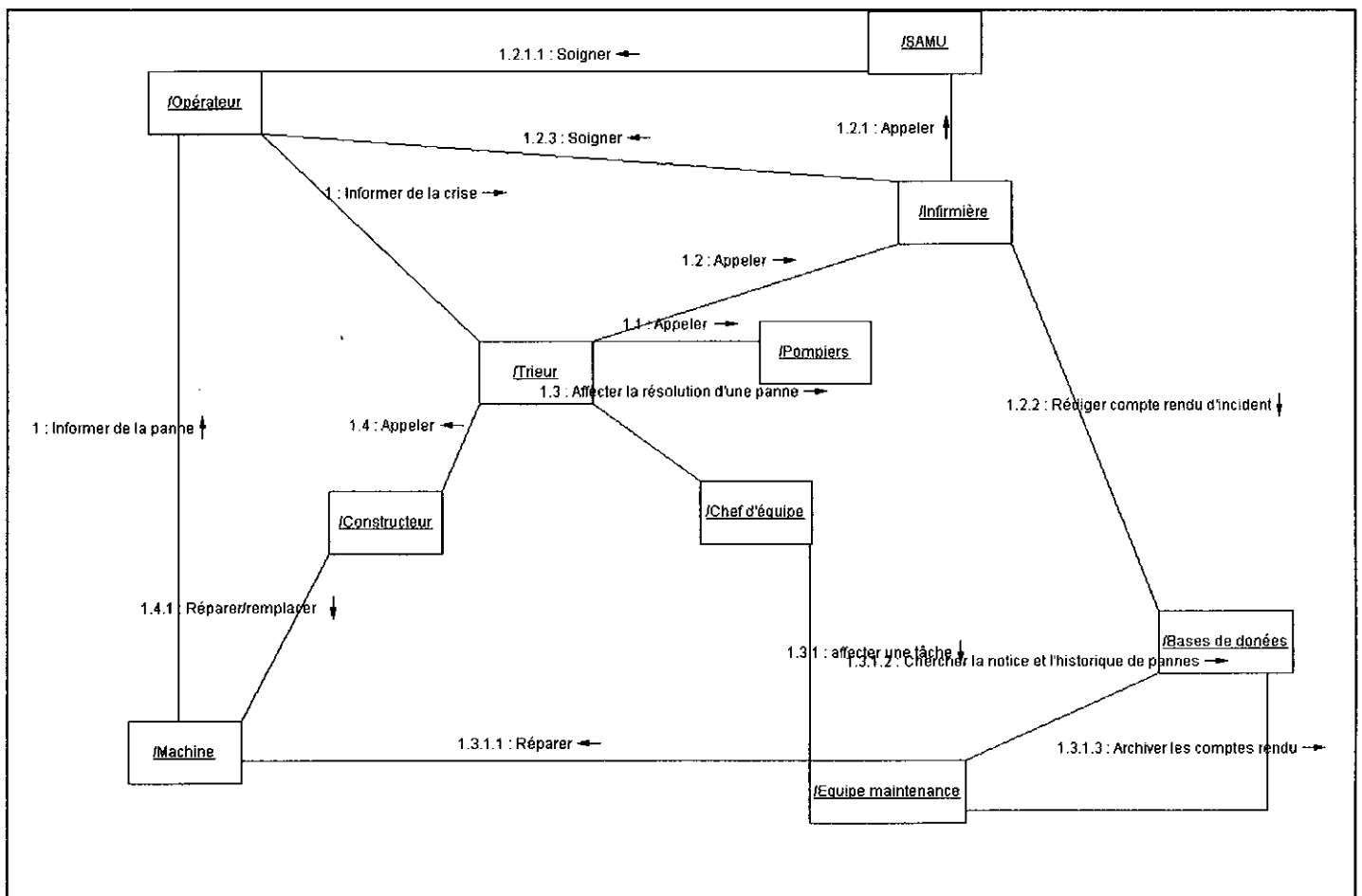


7

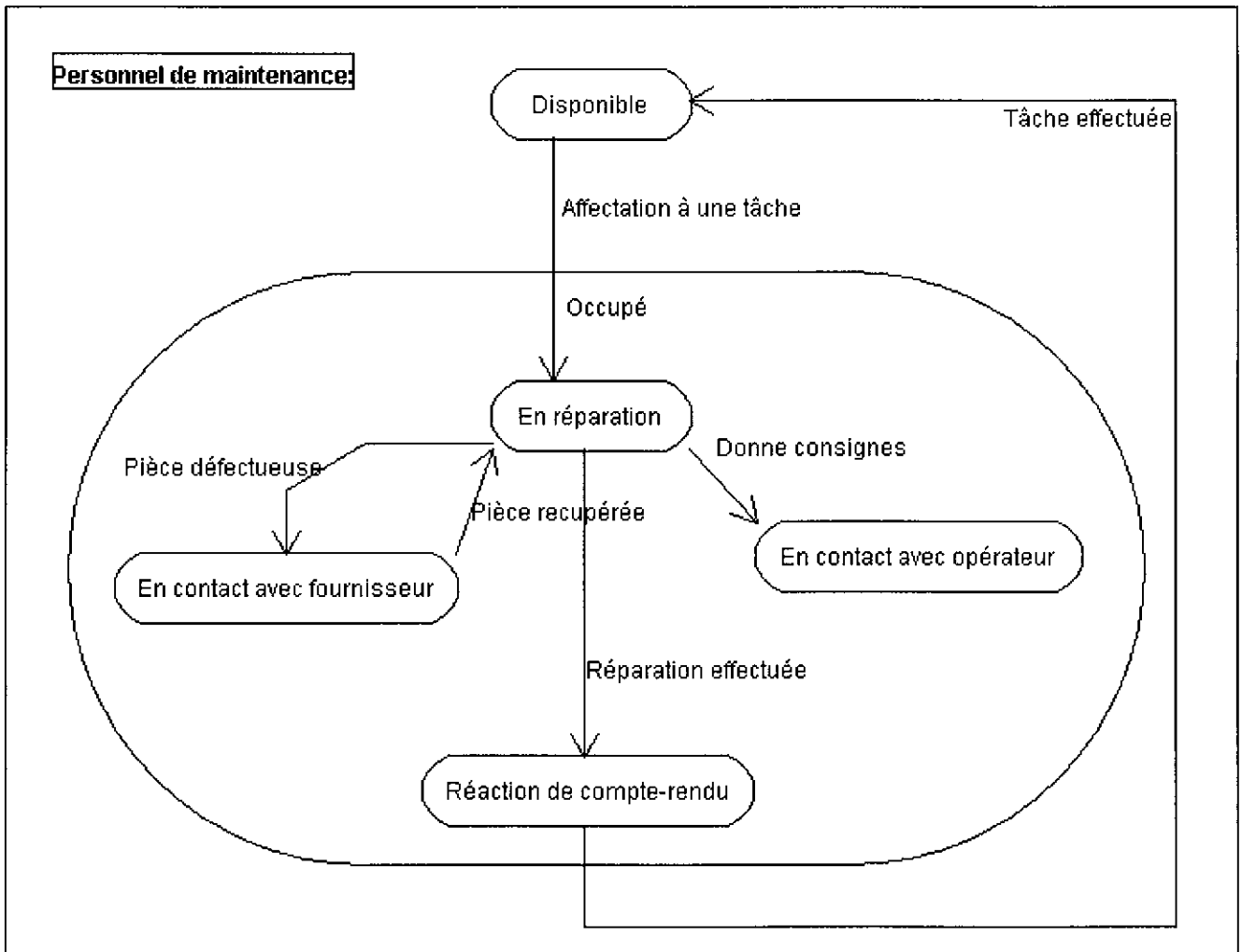
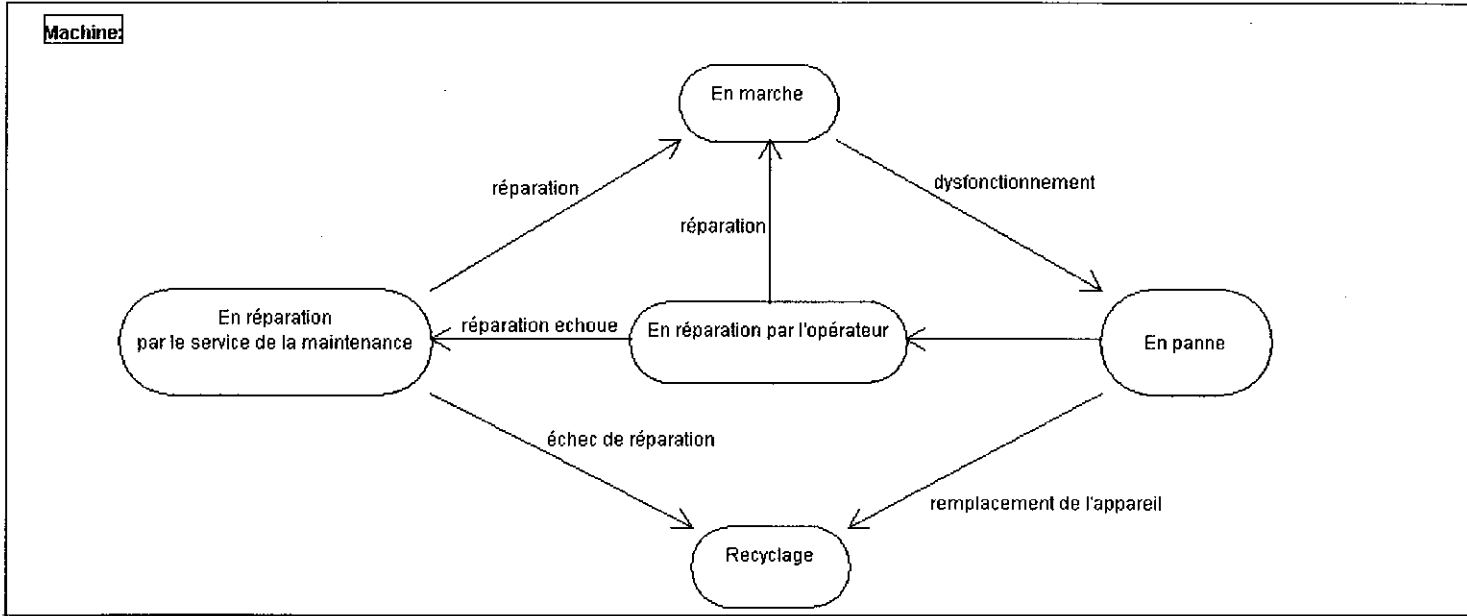




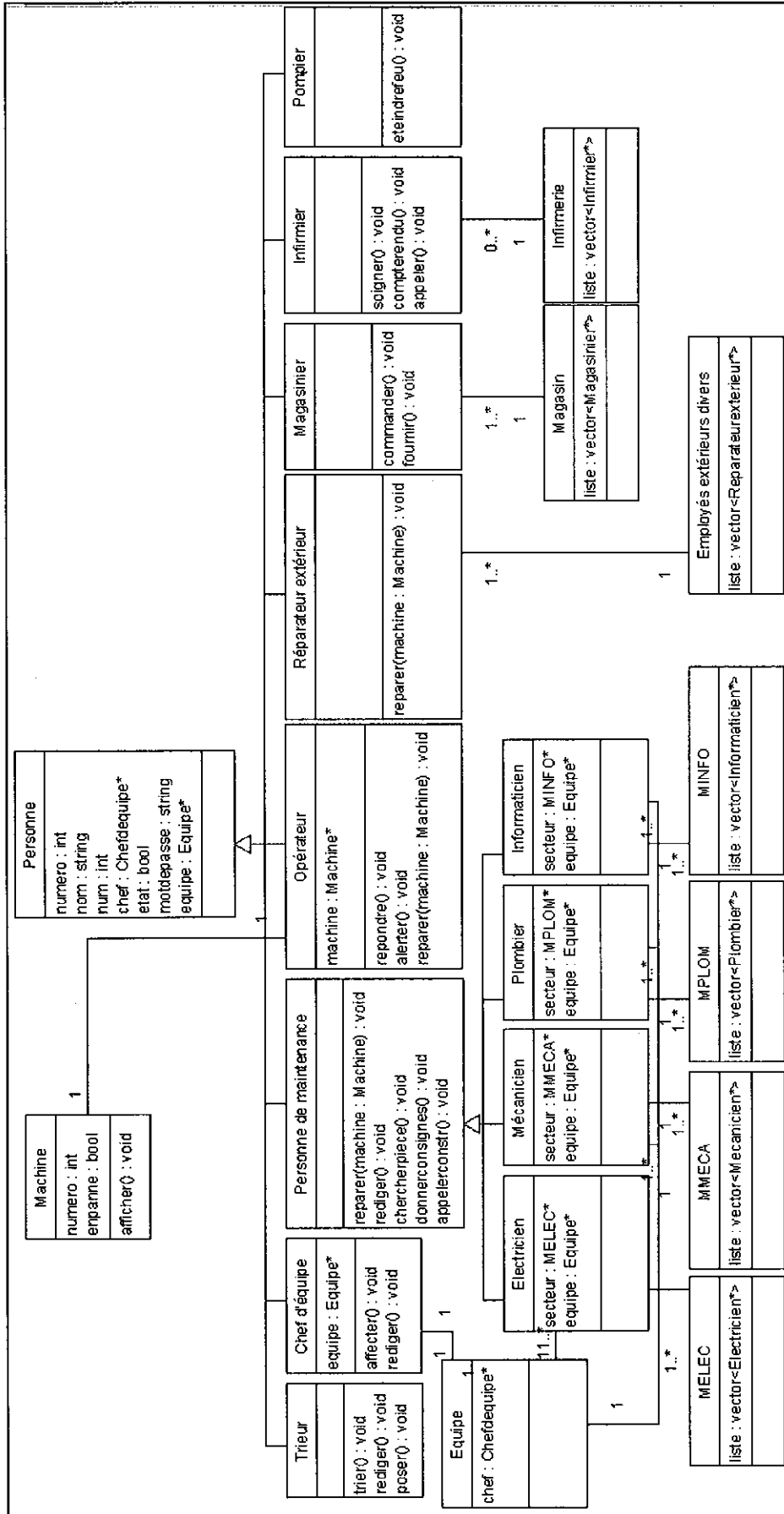
c. Diagramme de collaboration :



d. Diagrammes d'Etats :

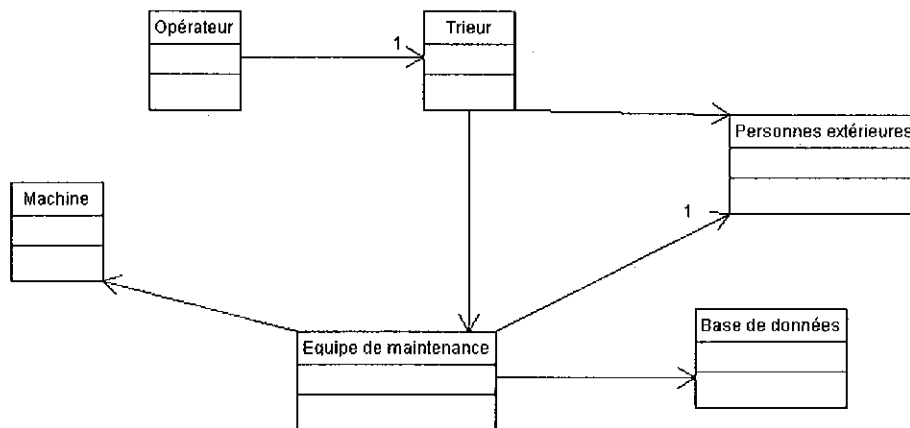


e. Diagramme de classes



### III/ Conception d'architecture :

#### *1. Diagramme d'architecture*



#### *2. Tests d'intégration*

##### Test sur le cas d'utilisation n°1

- L'opérateur simule une rupture de stock
- Il appelle le trieur pour commander la pièce (référence de la pièce et quantité)
- Le trieur commande la pièce au magasinier
- Le magasinier vérifie la disponibilité
- Il livre la pièce
- L'opérateur remplit le compte-rendu et l'archive dans la base de données

##### Test sur le cas d'utilisation n°2

- L'opérateur simule une panne
- Il arrête la machine
- Il appelle le trieur
- Le trieur lui pose des questions pour pouvoir commencer à remplir le compte-rendu et identifier le secteur concerné
- Le trieur identifie le bon secteur
- Le trieur appelle le bon chef d'équipe
- Le chef d'équipe coordonne son équipe
- Le technicien de maintenance récupère la documentation nécessaire (manuel d'utilisation et historique des pannes)
- Le technicien donne ses instructions de réparation à l'opérateur
- L'opérateur répare la machine
- Le technicien remplit le compte-rendu et l'archive dans la base de données
- Il prévient son chef d'équipe qu'il est libre

##### Test sur le cas d'utilisation n°3

- L'opérateur simule une panne
- Il arrête la machine
- Il appelle le trieur

#### Gestion de crise

- Le trieur lui pose des questions pour pouvoir commencer à remplir le compte-rendu et identifier le secteur concerné
- Le trieur identifie le bon secteur
- Le trieur appelle le bon chef d'équipe
- Le chef d'équipe coordonne son équipe
- Le technicien de maintenance récupère la documentation nécessaire (manuel d'utilisation et historique des pannes)
- Le technicien se déplace et répare la machine
- L'opérateur répare la machine
- Le technicien remplit le compte-rendu et l'archive dans la base de données
- Il prévient son chef d'équipe qu'il est libre

#### Test sur le cas d'utilisation n°4

- L'opérateur simule une panne
- Il arrête la machine
- Il appelle le trieur
- Le trieur lui pose des questions pour pouvoir commencer à remplir le compte-rendu et identifier le secteur concerné
- Le trieur identifie le bon secteur
- Le trieur appelle le bon chef d'équipe
- Le chef d'équipe coordonne son équipe
- Le technicien de maintenance récupère la documentation nécessaire (manuel d'utilisation et historique des pannes)
- Le technicien se déplace mais n'arrive pas à réparer la machine
- Le technicien appelle le fournisseur (il récupère le numéro dans la documentation)
- Le fournisseur se déplace sur place et répare la machine ou la remplace
- Le technicien remplit le compte-rendu et l'archive dans la base de données
- Il prévient son chef d'équipe qu'il est libre

#### Test sur le cas d'utilisation n°5

- L'opérateur simule un accident corporel
- Un de ses collègues appelle le trieur
- Le trieur demande des précisions
- L'opérateur le renseigne
- Le trieur informe l'infirmière
- L'infirmière appelle le SAMU
- Le SAMU intervient sur la personne
- L'infirmière rédige et archive un compte-rendu

#### Test sur le cas d'utilisation n°6

- L'opérateur simule un accident corporel de faible gravité
- Il appelle le trieur (ou un de ses collègues)
- Le trieur demande des précisions
- L'opérateur le renseigne
- Le trieur informe l'infirmière
- L'infirmière intervient sur la personne

- L'infirmière rédige et archive un compte-rendu

#### Test sur le cas d'utilisation n°7

- Un opérateur simule un incident grave dans l'usine (incendie, effondrement de la structure,...)
- Il appelle le trieur
- Le trieur demande des précisions
- L'opérateur donne des précisions
- Le trieur appelle les pompiers
- Les pompiers interviennent
- Les pompiers informent de la fin de l'intervention
- Le trieur appelle l'entreprise de déblaiement
- Les entreprises de déblaiement enlèvent les anciennes machines
- Les entreprises de déblaiement informent de la fin de l'intervention
- Le trieur commande les machines à remplacer
- Les fournisseurs installent les nouvelles machines
- Les fournisseurs informent de la fin de l'intervention
- Le trieur informe l'opérateur de la reprise possible de l'activité
- Le trieur archive le compte-rendu.

#### Test sur le cas d'utilisation n°8

- L'opérateur simule une panne
- Il arrête la machine
- Il appelle le trieur
- Le trieur lui pose des questions pour pouvoir commencer à remplir le compte-rendu et identifier le secteur concerné
- Le trieur identifie le bon secteur
- Le trieur appelle le bon chef d'équipe
- Le chef d'équipe coordonne son équipe
- Le technicien de maintenance récupère la documentation nécessaire (manuel d'utilisation et historique des pannes)
- Le technicien se déplace mais n'arrive pas à réparer la machine car il manque une pièce
- Le technicien commande la pièce au fournisseur
- Le fournisseur livre la pièce au magasinier
- Le magasinier livre la pièce au technicien
- Le technicien répare la machine
- Le technicien remplit le compte-rendu et l'archive dans la base de données
- Il prévient son chef d'équipe qu'il est libre.