

# E4D : ÉTUDE DE CAS

Durée : 5 heures

Coefficient : 5

## CAS UST

Ce sujet comporte 15 pages dont 5 pages d'annexes.  
Il est constitué de 5 dossiers qui peuvent être traités de façon indépendante.  
Le candidat est invité à vérifier qu'il est en possession d'un sujet complet.

### Matériels et documents autorisés :

- Lexique SQL sans commentaire ni exemple d'utilisation des instructions.
- Règle à dessiner les symboles informatiques.

**Aucune calculatrice n'est autorisée**

### Liste des annexes

- Annexe 1 : 1A - Extrait de dossier patient  
1B - Liste des codes médecins
- Annexe 2 : schéma partiel du réseau
- Annexe 3 : interface et algorithme de saisie d'une livraison
- Annexe 4 : descriptif des classes
- Annexe 5 : procédures événementielles de l'application sur l'ordinateur de poche

### Barème

Dossier 1 : Création des dossiers des patients en consultation externe	27 points
Dossier 2 : Référencement des produits à addiction	13 points
Dossier 3 : Administration du réseau du centre hospitalier	15 points
Dossier 4 : Gestion des médicaments	30 points
Dossier 5 : Installation de l'application	15 points
<b>Total</b>	<b>100 points</b>

CODE ÉPREUVE : ISE4D		EXAMEN : BREVET DE TECHNICIEN SUPÉRIEUR	SPÉCIALITÉ : INFORMATIQUE DE GESTION Option Développeur d'applications	
SESSION 2012	SUJET	ÉPREUVE : ÉTUDE DE CAS		
Durée : 5 h	Coefficient : 5		Code sujet : 12DA07N	Page : 1/15

## Présentation du contexte

Le centre hospitalier de la ville de M... est un important établissement de soins hébergeant de nombreuses spécialités. Au sein de ce centre hospitalier se trouve l'Unité de Soins en Toxicomanie appelée plus communément UST, qui dépend du secteur psychiatrique.

L'UST a pour mission principale de soigner les patients ayant une addiction (dépendance) envers des produits comme le tabac, l'alcool, ou encore les stupéfiants.

L'UST prend en charge :

- les consultations externes de patients suivis régulièrement et qui retournent à leur domicile à l'issue des visites ;
- l'hospitalisation des patients pour des durées plus ou moins longues dans le cadre notamment des cures de sevrage.

<b>DOSSIER 1</b>	<b>Création des dossiers des patients en consultation externe</b>
------------------	---

*Documents à utiliser : annexes 1A, 1B*

L'UST a décidé de restructurer la gestion des dossiers de ses patients en consultation externe (ceux qui ne sont pas hospitalisés). Pour cela, elle souhaite procéder à une étude approfondie des informations indispensables à la constitution et au suivi des dossiers.

### Les équipes

Pour une bonne organisation au sein de l'unité, le personnel infirmier est regroupé en équipes. Chaque équipe se réunit régulièrement pour faire le point sur les patients dont elle est responsable. La date et l'heure de la prochaine réunion de chaque équipe doivent être mémorisées.

### Constitution du dossier

L'infirmier qui accueille un nouveau patient remplit un dossier présenté en **annexe 1A**. Il y consigne les informations concernant ce patient et établit un premier diagnostic (correspondant au commentaire du dossier). Il sera responsable du patient jusqu'à la clôture du dossier.

Un dossier est repéré à l'aide d'un numéro séquentiel attribué par année. Le dossier patient dont un extrait est présenté en **annexe 1A** est donc le 52<sup>ème</sup> dossier de l'année 2012.

La référence au médecin traitant (si le patient en a un) est indiquée à l'aide d'un code à partir d'une liste tenue à jour (voir **annexe 1B**).

À la fin de l'entretien, un premier rendez-vous est fixé.

## Suivi du dossier

Chaque rendez-vous avec un patient est pris en charge par un infirmier de l'UST, qui n'est pas forcément l'infirmier responsable du patient.

À chaque rendez-vous, deux cas peuvent se produire :

1. Le patient ne vient pas au rendez-vous. L'infirmier enregistre alors son absence et le contacte pour fixer un autre rendez-vous.
2. Le patient se présente au rendez-vous. Dans ce cas, l'infirmier enregistre sa présence et commence l'entretien.

## L'entretien

Lors d'un entretien, l'infirmier questionne le patient et complète son dossier en fonction des informations qu'il obtient concernant les produits consommés d'une part, les traitements de substitution suivis d'autre part. Il n'est pas rare que ces informations ne soient fournies par le patient qu'après plusieurs rendez-vous en fonction de son niveau de confiance.

Le fait que le patient soit ou non motivé est évalué par l'infirmier. On conserve l'historique de la motivation du patient constatée à chaque rendez-vous.

La clôture du dossier intervient lorsque le patient est considéré comme guéri ou qu'il renonce à se soigner. La date de clôture du dossier est alors enregistrée.

Un patient dont le dossier a été clos pourra à nouveau être admis dans l'unité. Dans ce cas, un nouveau dossier sera créé.

## Les produits consommés

Il arrive fréquemment qu'un patient consomme un produit nocif, arrête cette consommation pendant quelques temps, puis la reprenne.

La consommation de chaque produit nocif doit être consignée : pour chaque période, il faut enregistrer la date de début et éventuellement la date de fin de la prise de produit, la fréquence des prises et la dose moyenne absorbée par prise.

## Les traitements de substitution

Un traitement de substitution consiste à prescrire un ou plusieurs médicaments de manière à aider le patient à cesser la prise de produits. Chaque traitement est prescrit par un médecin (la liste des médecins est présentée en **annexe 1B**) et possède une date de prescription.

Pour chaque médicament prescrit dans le cadre d'un traitement, on mémorise la date de début de prise, la durée prescrite et la posologie.

Travail à faire	
1.1	Présenter le schéma conceptuel de données correspondant au suivi des consultations externes.

Le personnel médical utilise une base de données référençant les produits pouvant entraîner des addictions. Cette base permet de connaître pour chaque produit toxique les risques (dépendance psychique, maladie cardiovasculaire, etc.) et le niveau de dangerosité associé.

Le schéma relationnel de cette base est le suivant :

**PRODUIT** (num, nom)

*num : clé primaire*

**RISQUE** (code, libelléR)

*code : clé primaire*

**NIVEAU** (num, libellé)

*num : clé primaire.*

**DANGEROUSITÉ** (numProduit, codeRisque, numNiveau)

*numProduit, codeRisque : clé primaire*

*numProduit : clé étrangère en référence à num de PRODUIT*

*codeRisque : clé étrangère en référence à code de RISQUE*

*numNiveau : clé étrangère en référence à num de NIVEAU*

Exemple : DANGEROUSITÉ(12, 3, 2) et DANGEROUSITÉ(12, 7, 3) signifient que le produit tabac (numProduit = 12) présente un risque de dépendance psychique (codeRisque = 3) de niveau « élevé » (numNiveau = 2) et un risque de maladie cardiovasculaire (codeRisque = 7) de niveau « très élevé » (numNiveau = 3).

### Travail à faire

- 2.1 Écrire en SQL les requêtes permettant de réaliser les opérations suivantes :
- A. Afficher les noms des produits présentant un risque de "Dépendance psychique". La liste doit être triée alphabétiquement.
  - B. Afficher les niveaux de dangerosité pour lesquels le nombre de produits est supérieur à dix (numéro, libellé et nombre de produits).
  - C. Afficher le numéro et le nom des produits présents dans la base pour lesquels les informations concernant la dangerosité ne sont pas encore renseignées.

La structure de la base de données doit être modifiée pour permettre au personnel médical de classer les produits par catégorie. Les modifications sont les suivantes :

→ Création d'une nouvelle table CATEGORIE :

**CATÉGORIE** (code, nom)

*code : clé primaire*

→ Modification de la table PRODUIT :

**PRODUIT** (num, nom, **codeCatégorie**)

*num : clé primaire*

**codeCatégorie : clé étrangère en référence à code de CATEGORIE**

Table	Colonne	Type de données
CATEGORIE	code	Entier
CATEGORIE	nom	Chaîne de caractères de longueur variable (30 caractères maximum)

### Travail à faire

- 2.2 Écrire les requêtes SQL permettant d'effectuer les modifications de structure de la base de données (incluant les contraintes) en prenant soin d'écrire les requêtes dans l'ordre où elles doivent être exécutées.
- 2.3 Écrire la requête SQL permettant de n'attribuer que les droits de lecture, d'insertion, de modification et de suppression de lignes sur la table CATEGORIE à l'utilisateur nommé "application".

Le réseau actuel est basé sur une architecture Ethernet 100 Mbit/s. Une connexion SDSL est disponible.

*Remarque : le schéma de ce réseau présenté en **annexe 2** ne prend pas en compte les éventuelles modifications qui résulteront de la mise en exploitation du logiciel présenté dans le dossier 4.*

Le serveur HTTP se trouvant dans la DMZ abrite le site *web* du centre hospitalier et doit être accessible depuis l'extérieur via Internet.

Travail à faire	
3.1	Indiquer pourquoi, quelles que soient les configurations mises en place, il est impossible d'accéder depuis l'extérieur via Internet au site <i>web</i> du centre hospitalier par l'URL <a href="http://192.168.3.254">http://192.168.3.254</a> .

Les postes de la pharmacie centrale doivent pouvoir communiquer avec le serveur *proxy* de la DMZ afin d'accéder à Internet. Les tables de routage des deux routeurs sont les suivantes :

#### Table de routage pour le routeur R1

Numéro de ligne	Réseau	Routeur (Passerelle)	Interface
<b>R1-1</b>	192.168.2.0/25	192.168.1.253	192.168.1.254
<b>R1-2</b>	192.168.1.0/24	192.168.1.254	192.168.1.254
<b>R1-3</b>	192.168.3.0/24	192.168.3.254	192.168.3.254
<b>R1-4</b>	0.0.0.0/0	82.127.150.99	82.127.150.98

#### Table de routage pour le routeur R2

Numéro de ligne	Réseau	Routeur (Passerelle)	Interface
<b>R2-1</b>	192.168.2.0/25	192.168.2.126	192.168.2.126
<b>R2-2</b>	0.0.0.0/0	192.168.1.254	192.168.1.253

La route par défaut sur les routeurs s'exprime à l'aide du numéro de réseau 0.0.0.0/0

L'informaticien en charge du réseau effectue le test suivant : à partir du poste de la pharmacie centrale d'adresse 192.168.2.10, il exécute la commande "ping 192.168.3.1". Cette commande réussit.

Travail à faire	
3.2	Indiquer les lignes des tables de routage utilisées pour la réalisation de cette commande (trafic aller et retour).

Pour des raisons de sécurité, toutes les requêtes provenant des postes du centre hospitalier destinées à des serveurs HTTP (port 80) situés sur Internet sont redirigées automatiquement vers le serveur *proxy* sur le port 3128. La règle suivante a été écrite sur l'interface 192.168.1.254.

Num	Interface	Protocole	Adresse destination	port destination	adresse de redirection	port de redirection
1	192.168.1.254	TCP	*	80	192.168.3.1	3128

Cette règle signifie qu'en entrée de l'interface 192.168.1.254, si le port de destination est 80, le paquet sera redirigé vers l'adresse 192.168.3.1, port 3128.

Le serveur *web* écoute le port 80. Des tests effectués depuis l'extérieur via Internet montrent que l'accès au site *web* du centre hospitalier à partir d'un navigateur par l'URL <http://82.127.150.98> échoue.

Travail à faire	
3.3	En s'inspirant de la règle présentée ci-dessus, écrire la règle qui va permettre d'accéder depuis l'extérieur via Internet au site <i>web</i> du centre hospitalier.

Les postes des services médicaux et le serveur médical ne doivent pas communiquer avec le reste du réseau.

Travail à faire	
3.4	Expliquer comment le plan d'adressage permet de respecter cette contrainte.

Certaines addictions, avec l'accord du patient, sont traitées par son hospitalisation dans le centre.

Afin de réduire le coût des médicaments, le directeur a demandé à la pharmacie centrale de prescrire systématiquement le traitement le moins coûteux. La pharmacie s'approvisionne pour cela auprès de différents laboratoires et négocie les commandes. Une livraison concerne toujours un seul médicament ; la livraison se fait par boîte et les médicaments sont ensuite déconditionnés. Le stock est donc exprimé en nombre de comprimés.

À chaque livraison, le coût du comprimé est recalculé grâce à la méthode du coût unitaire moyen pondéré et le stock de médicaments est mis à jour.

*Exemple : Le stock restant de Butexu est de 50 comprimés pour un coût moyen de 40 centimes le comprimé. Le pharmacien réceptionne une livraison de Butexu de 100 boîtes de 7 comprimés chacune pour un coût total de livraison de 400 €. Le nouveau coût unitaire moyen pondéré (CUMP) de 56 centimes est obtenu en divisant la valeur totale du stock (valeur du stock restant + valeur du stock entrant) par la quantité totale de comprimés (quantité en stock + quantité livrée). Après la livraison, le stock est de 750 comprimés.*

Le pharmacien qui réceptionne les livraisons utilisera une application dont le formulaire est présenté en **annexe 3**. Une première version de la procédure d'enregistrement d'une livraison produit une erreur. Lors de la compilation de l'instruction de la dernière ligne de la méthode `leMedicament.majCump(qte, condi, cout)` décrite en **annexe 3**, le message d'erreur « Pas de définition pour cette méthode » apparaît.

Travail à faire	
4.1	En observant la classe Medicament de l' <b>annexe 4</b> , expliquer la cause de l'apparition de ce message alors que la méthode <code>majCump()</code> est bien définie dans la classe Medicament.
4.2	Écrire la méthode <code>majCump()</code> de la classe Medicament permettant de calculer et modifier le CUMP (le nouveau coût moyen d'un comprimé).
4.3	Écrire la méthode <code>ajoutStock()</code> de la classe Medicament permettant de calculer et modifier le CUMP et de mettre à jour le stock.

La méthode de classe (statique) `ajouterLivraison()` de la classe Pharmacie permet d'ajouter la livraison à la collection de livraisons existantes et de faire les modifications nécessaires (calcul du CUMP et mise à jour du stock) pour l'objet de la classe Medicament livré.

Travail à faire	
4.4	Écrire la méthode <code>ajouterLivraison()</code> de la classe Pharmacie.

Pour un même traitement, il existe plusieurs médicaments sur le marché, ayant la même efficacité puisqu'utilisant la même molécule.

*Exemple : pour la molécule buprénorphine (traitement des dépendances à certaines drogues), trois médicaments sont disponibles à la pharmacie : Buprénorphine arrow (générique), Butexu et Gésictem.*

Travail à faire	
4.5	Écrire la méthode <code>donneLePlusEconomique()</code> de la classe Molecule. On pourra utiliser une constante nommée <code>MAXVALUE</code> représentant la plus grande valeur réelle utilisable.

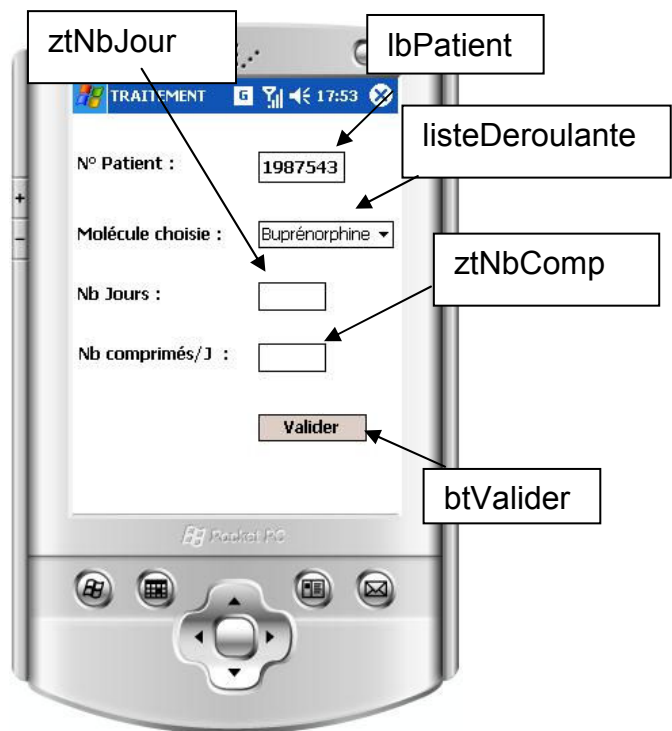
Chaque matin, un médecin passe dans la chambre de chaque patient pour suivre l'évolution de sa pathologie et éventuellement changer son traitement. Le directeur de l'hôpital a demandé au service informatique de développer une application pour PDA ("Personal Digital Assistant" : ordinateur de poche) afin que le médecin puisse saisir la molécule à prescrire lorsque le traitement est administré sous forme de comprimé.

L'interface proposée par le service informatique est la suivante :

L'identification du malade a été réalisée grâce à la lecture par le PDA du code-barre du dossier.

Le médecin saisit la molécule (parmi l'ensemble des molécules proposées dans une liste déroulante), la durée du traitement en nombre de jours et la quantité de comprimés à prendre par jour. Le médecin valide ensuite le traitement.

Pour mettre en œuvre cette application mobile, vous allez utiliser un service *web*. Il s'agit d'une classe hébergée sur un serveur *web*. Chaque méthode de cette classe attend des paramètres fournis par l'application cliente et rend à celui-ci un résultat.



La classe ServicesPharmacie, présentée en **annexe 4**, constitue la classe de ce service *web*. Un exemple d'appel du service *web* est présenté en **annexe 5** dans la procédure *traitement\_chargement()*.

Travail à faire	
4.6	Écrire la méthode <i>ajoutTraitementBase()</i> de la classe ServicesPharmacie décrite en <b>annexe 4</b> .
4.7	Écrire la procédure événementielle <i>btValider_Click()</i> de l' <b>annexe 5</b> . N'indiquer sur la copie que la partie à compléter.



La création de nouveaux services *web* va nécessiter un renfort temporaire de l'équipe chargée du développement. Le responsable informatique envisage soit de faire appel à une société de services en ingénierie informatique (SSII) externe, soit de former un des informaticiens du centre hospitalier pour assurer le développement en interne.

Il dispose des éléments chiffrés suivants :

➔ **pour le développement en interne**

- deux formations nécessaires :
  - "Développement *web*" : 2 jours – 1 150 euros ;
  - "Développer des services *web*" : 4 jours – 2 300 euros ;
- coût salarial de l'informaticien du centre hospitalier : 200 euros par jour ouvrable, qu'il soit en formation ou en développement ;
- indemnité par jour de formation (repas et déplacement) payée à l'informaticien du centre hospitalier : 25 euros.

Le stage s'effectue durant les jours de travail de l'informaticien ; son coût salarial durant cette période doit être pris en compte.

➔ **pour le développement en externe**

- prix facturé à la journée par la SSII : 400 euros.

La sous-traitance semble, a priori, préférable aux opérations de formation puis de développement interne.

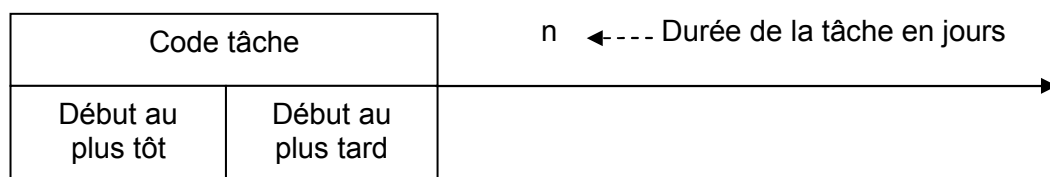
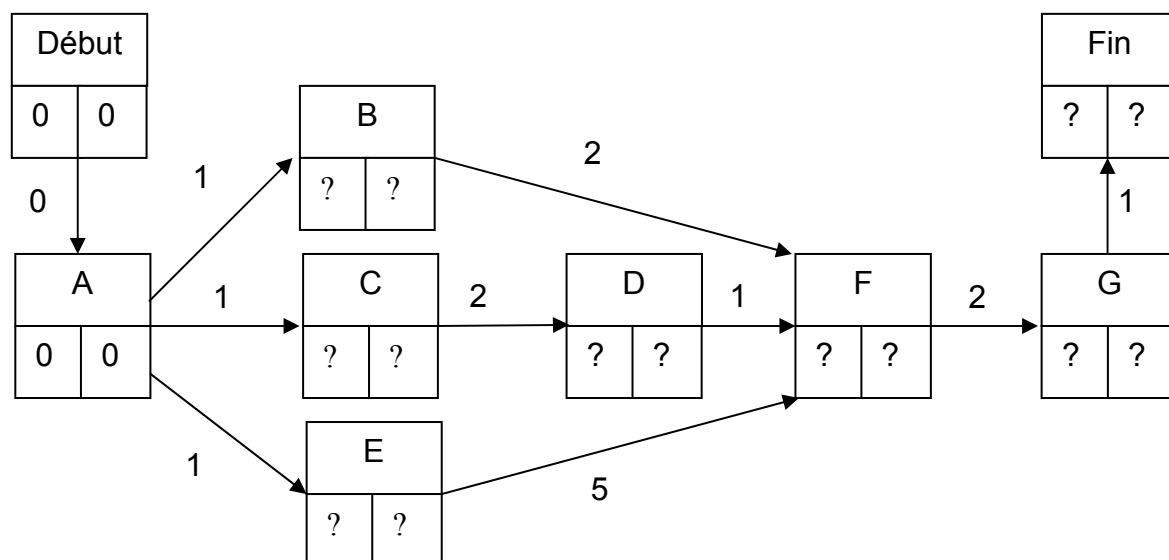
Travail à faire	
5.1	En <i>justifiant par calcul</i> votre réponse, indiquer le nombre de jours de développement à partir duquel il est plus rentable de former l'informaticien du centre hospitalier plutôt que de faire appel à une société de services.
5.2	Indiquer en quoi le coût ne peut pas être le seul critère de décision pour le responsable informatique.

L'utilisation par les médecins de l'application développée dans le dossier 4 nécessite auparavant la réalisation des tâches suivantes par le service informatique :

Code tâche	Descriptif de la tâche
A	Réception du matériel (PDA, serveur dédié, etc.) et contrôle
B	Configuration des PDA
C	Installation et configuration du serveur HTTP
D	Installation des services <i>web</i>
E	Modification et configuration du réseau ( <i>Wi-Fi</i> , routeur, serveur dédié, etc.)
F	Test d'intégration
G	Formation des utilisateurs en configuration réelle

Un diagramme MPM a été réalisé avec les contraintes suivantes :

- une tâche n'est effectuée que par une seule personne ;
- le nombre de personnes disponibles est toujours suffisant et elles disposent des compétences nécessaires pour l'ensemble des tâches.



Travail à faire	
5.3	Indiquer les codes des tâches faisant partie du chemin critique.

La première tâche débute le lundi 4 juin 2012. De nombreux projets étant en cours à ce moment, le responsable informatique ne dispose finalement que de :

- Marie, disponible pour le mois de juin et disposant de toutes les compétences nécessaires ;
- Julien, disponible uniquement du lundi 4 au vendredi 8 juin et ne possédant pas les compétences pour installer et configurer le serveur HTTP.

La réception du matériel (PDA, etc.) et le contrôle doivent être réalisés par Marie qui a en charge la réception de tout le matériel informatique au sein du centre hospitalier.

Le personnel travaille cinq jours par semaine (du lundi au vendredi). Le responsable affecte toujours une tâche à une seule personne.

Modèle de planning utilisé par le responsable informatique :

	Juin							
	Mardi 5	Mercredi 6	Jeudi 7	Vendredi 8	Samedi 9	Dimanche 10	Lundi 11	...
Aïcha	L						L	...
Samuel	K	M					N	...

Dans ce modèle K, L, M et N représentent des tâches.

Le planning à réaliser doit prendre en compte l'ensemble des contraintes et permettre la réalisation de l'ensemble des tâches dans le délai le plus court.

Travail à faire	
5.4	Proposer un planning respectant le modèle présenté en affectant les tâches aux deux informaticiens Marie et Julien. En déduire à quelle date l'ensemble des tâches sera terminé.

## Annexe 1

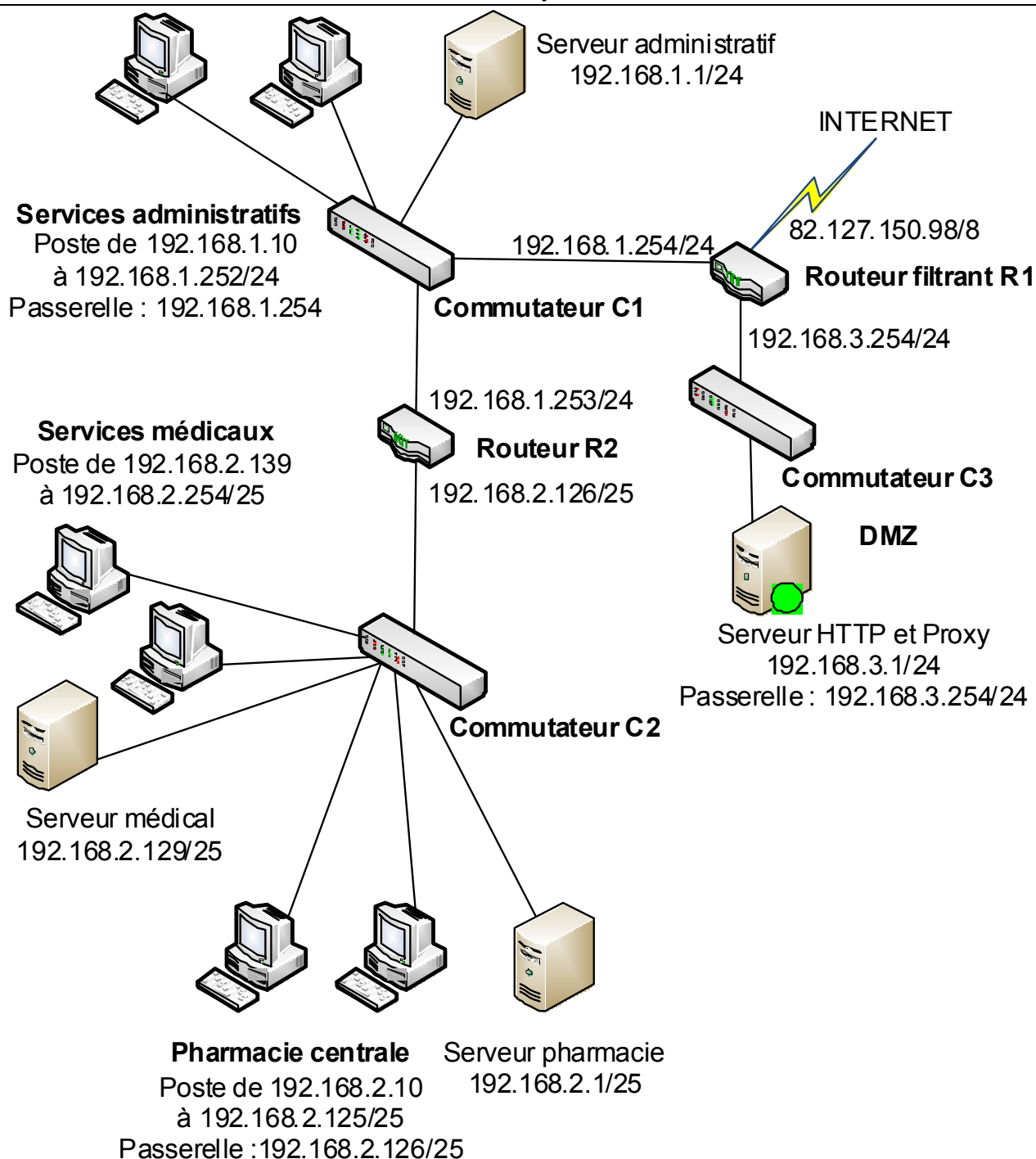
### Annexe 1A - Extrait de dossier patient

<b>DOSSIER</b>													
<b>N° : 52</b>			<b>Année : 2012</b>										
<b>INFIRMIER</b>	<b>Code : 502</b>	<b>Nom : PERRET</b>	<b>Prénom : Inès</b>										
<b>PRISE EN CHARGE</b>	<b>Date de début : 05/04/2012</b>	<b>Date de fin :</b>											
<b>PATIENT</b>	<b>Nom : PEUDAIDE</b>		<b>Prénom : Justin</b>										
<b>Date naissance :</b> 16/10/1961 <b>Adresse :</b> 14, rue des Lilas 75020 PARIS  <b>Téléphone :</b> 01 42 03 14 15													
<b>Situation de famille :</b> <div style="display: flex; justify-content: space-between;"> <div> <input type="checkbox"/> Célibataire  <input type="checkbox"/> Marié (e)  <input checked="" type="checkbox"/> Veuf (ve)           </div> <div> <input type="checkbox"/> Séparé (e)  <input type="checkbox"/> PACS  <input type="checkbox"/> Divorcé (e)           </div> <div> <input type="checkbox"/> Concubinage           </div> </div>													
<b>Code du médecin traitant :</b> 032													
<b>Commentaire :</b> Conduite nocive, voire installation d'une dépendance. Patient qui semble motivé à travailler une prise en charge.													
<b>Historique des rendez-vous</b>  <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">Date</th> <th style="width: 80%;">Présence (O/N)</th> </tr> </thead> <tbody> <tr> <td>10/04/2012</td> <td>N</td> </tr> <tr> <td>16/04/2012</td> <td>O</td> </tr> <tr> <td>23/04/2012</td> <td>O</td> </tr> <tr> <td>...</td> <td></td> </tr> </tbody> </table>				Date	Présence (O/N)	10/04/2012	N	16/04/2012	O	23/04/2012	O	...	
Date	Présence (O/N)												
10/04/2012	N												
16/04/2012	O												
23/04/2012	O												
...													

### Annexe 1B - Liste des codes médecins

Code	Nom	Prénom	Adresse rue Cabinet	CP Cabinet	Ville Cabinet	Téléphone
001	CARRE	Georges	9, rue des acacias	75009	PARIS	0123456778
002	MOMALLE	Jean	62, bd Leclerc	75009	PARIS	0198765432
....	....	....	....	....	....	....
032	AZOUZ	Djamila	18, rue des haies	75020	PARIS	0143727297
....	....	....	....	....	....	....

## Annexe 2 - Schéma partiel du réseau

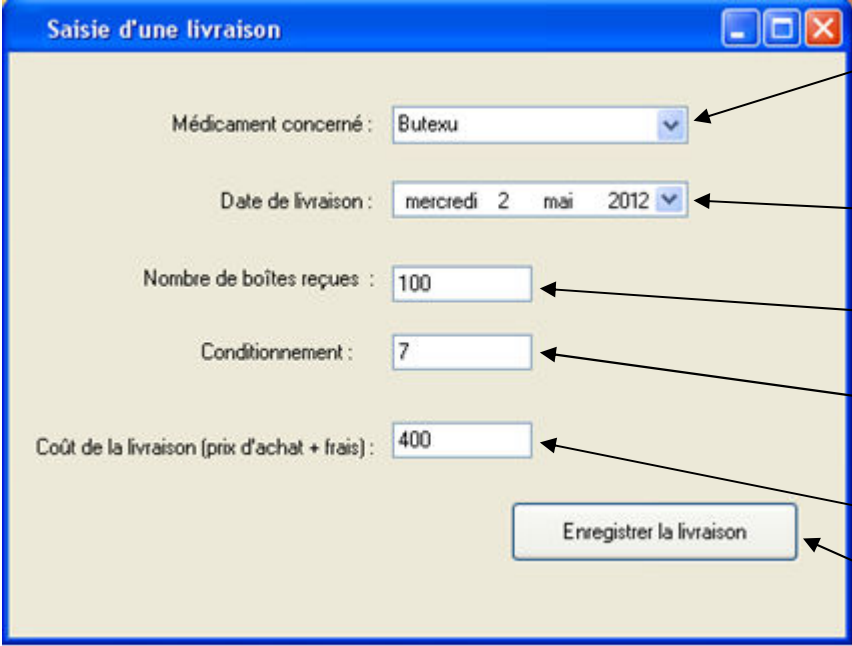


**Plan d'adressage : quatre réseaux IP différents sont gérés.**

- **192.168.1.x** : réseau administratif - Adresse réseau : 192.168.1.0/24
- **192.168.2.x** : réseau médical : il comprend deux sous-réseaux.
  - Pharmacie centrale - Adresse réseau : 192.168.2.0/25
  - Services médicaux - Adresse réseau : 192.168.2.128/25
- **192.168.3.x** : DMZ - Adresse réseau : 192.168.3.0/24

**Remarque :** L'interface 192.168.2.126/25 du routeur R2 possède la même adresse réseau que les postes de la pharmacie centrale.

## Annexe 3 - Interface et algorithme de saisie d'une livraison

Interface	Noms des contrôles
	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">listeDesMedicaments</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">listeDateLivraison</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">zoneDeTexteQte</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">zoneDeTexteCondi</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">zoneDeTexteCout</div> <div style="border: 1px solid black; padding: 5px;">boutonValidation</div>

### Procédure événementielle boutonValidation\_Click()

*/\* déclaration des variables locales \*/*

nomMedicament : chaîne  
 dateLiv : Date  
 qte : entier  
 condi : entier  
 cout : réel  
 leMedicament : Medicament  
 unObjetLivraison : Livraison

*/\* récupération du contenu des zones \*/*

nomMedicament ← listeDesMedicaments.valeur  
 dateLiv ← listeDateLivraison.valeur  
 qte ← zoneDeTexteQte.valeur  
 condi ← zoneDeTexteCondi.valeur  
 cout ← zoneDeTexteCout.valeur

leMedicament ← Pharmacie.trouverMedicament(nomMedicament)

*/\* instanciation de l'objet LIVRAISON et ajout à la collection \*/*

unObjetLivraison = new Livraison (dateLiv, qte, condi, cout, leMedicament)  
 Pharmacie.ajouterLivraison(unObjetLivraison)

*/\* mise à jour du cump \*/*

**leMedicament.majCump(qte, condi, cout)**

Provoque un message d'erreur à la compilation :  
« Pas de définition pour cette méthode »

## Annexe 4 - Descriptif des classes

### Classe Medicament

Privé

nom : chaîne  
stock : entier // quantité de comprimés en stock  
cump : réel // coût moyen d'un comprimé en stock  
**procédure majCump(qté : entier, condit : entier, coût : réel)**  
// calcule et affecte le nouveau CUMP lors d'une livraison, ne modifie pas le stock

Public

Medicament(...) // constructeur  
fonction getNom() : chaîne  
fonction getCump() : réel  
fonction getStock() : entier  
**procédure ajoutStock(qté : entier, condit : entier, coût : réel)**  
// appel de la méthode majCump et modifie le stock

### Classe Livraison

Privé

dateLivraison : Date // date de la livraison  
qte : entier // quantité de boîte livrées  
condit : entier // nombre de comprimés par boîte  
coût : réel // coût d'achat de la livraison  
leMedicament : Medicament // objet Medicament concerné par la livraison

Public

Livraison(d : Date, q : entier, c : entier, coût : réel, leM : Medicament) // constructeur  
fonction getCout() : réel // retourne le coût d'achat de la livraison  
fonction getQte() : entier // retourne la quantité de boîtes livrées  
fonction getCondit() : entier // retourne le nombre de comprimés par boîte  
fonction getLeMedicament() : Medicament // retourne le médicament livré

### Classe Molecule

Privé

nom : chaîne  
LesMedicaments : collection de Medicament

Public

Molecule(...) // constructeur  
**Fonction donneLePlusEconomique(qte : entier) : Medicament**  
// renvoie l'objet Medicament dont le CUMP est le plus petit, et dont le stock est suffisant pour un  
// traitement (stock >= qte) ou renvoie NULL si aucun médicament n'a un stock suffisant.

### Classe Pharmacie // Toutes les variables et méthodes sont statiques c'est-à-dire à portée "classe"

Privé

lesMolecules : collection de Molecule  
lesLivraisons : collection de Livraison

Public

**procédure ajouterLivraison(uneLivraison : Livraison)**  
fonction trouverMedicament(nom : chaîne) : Medicament  
// renvoie l'objet Medicament correspondant au nom passé en paramètre.  
Procédure enregistrerTraitementBase(noPatient: entier, nomMedicament : chaîne, qte : entier)  
// enregistre la quantité de médicaments nécessaire pour le traitement du patient  
// dans la base de données.

**Classe ServicesPharmacie // classe de SERVICE WEB**

Privé

fonction trouverMolecule(nom : chaîne) : Molecule  
*// renvoie l'objet Molecule correspondant au nom passé en paramètre.*

Public

ServicesPharmacie (...) *// constructeur*

fonction listeNomMolecules() : tableau de chaîne

**fonction ajoutTraitementBase(noPatient : entier, nomMolecule : chaîne,  
nbJour : entier, nbComprime : entier) : booléen**

*// Recherche le médicament le moins cher et en stock correspondant au nom de la molécule transmis  
// en paramètre, puis enregistre dans la base de données le traitement du malade (en utilisant la classe  
// Pharmacie) et retourne vrai. Retourne faux si aucun médicament n'a été trouvé.*

**Classe Collection de <nom de la classe>**

Public

fonction cardinal() : entier *// renvoie le nombre d'éléments de la collection*

fonction extraireObjet(unIndex : Entier) : Objet de la classe

*// Retourne l'objet d'index unIndex. Le premier élément se trouve à l'index 1*

procédure ajouter(unObjet : Objet de la classe) *// Ajoute un objet à la collection*

*Pour instancier une collection :*

*uneCollection : Collection de <classe>*

*uneCollection ← new Collection() de <classe>*

*On peut aussi parcourir par itération les éléments d'un objet Collection*

*Pour chaque <objet> de <collection>*

*// instructions avec <objet>*

*FinPour*

**Annexe 5 - Procédures événementielles de l'application sur l'ordinateur de poche****Classe FormPda // Formulaire affiché sur le PDA**

Privé

scePharma : ServicesPharmacie

procédure événementielle traitement\_chargement()

début

tabNomMol : tableau de chaîne

*// au chargement, initialisation de l'attribut scePharma*

scePharma ← new ServicesPharmacie()

tabNomMol ← scePharma.listeNomMolecules()

*// Alimentation de la liste déroulante du PDA avec le contenu du tableau*

listeDéroulante.ajouter(tabNomMol)

fin

**procédure événementielle btValider\_Click()**

début

leNbJours, leNbComprimés : entier

leNomMolecule : chaîne

*// récupération du nom de la molécule se trouvant dans la liste déroulante du PDA.*

leNomMolecule ← listeDéroulante.valeur

*// récupération du nombre de jours se trouvant dans la zone de texte du PDA.*

leNbJours ← ztNbJour.valeur

*// récupération du nombre de comprimés se trouvant dans la zone de texte du PDA.*

leNbComprimés ← ztNbComp.valeur

*// récupération du numéro de patient se trouvant dans le label du PDA*

leNoPatient ← lbPatient.valeur

**...à compléter ...**

fin

procédure afficherMessage(message : chaîne)

*// affiche le message passé en paramètre sur l'écran du PDA.*