

Projet intégrateur 420-DA3-AS

Phase 1: Cahier de charges, analyse et séparation générale des tâches

Rappel

Le projet est divisé en 3 phases:

- Formation de l'équipe (semaines 1 à 3)
- Analyse: analyse du domaine, des cas d'utilisation et rapport d'analyse (semaines 4 à 7)
 - Analyse du domaine (semaine 4)
 - Analyse des cas d'utilisation (semaines 5-6)
 - Rapport d'analyse (semaine 7)
- Réalisation: production du code, revue et tests de qualité (semaines 8 à 15)
 - Réalisation : classes du domaine (entités) (semaine 8)
 - Réalisation : classe de contexte et configuration de base des entités avec *ModelBuilder* (semaine 9)
 - Réalisation : configuration des associations entre entités (semaine 10)
 - Réalisation : classes d'accès aux données (semaine 11)
 - Réalisation : classes de présentation (semaine 12)
 - Réalisation : classes métier (semaine 13)
 - Réalisation : intégration et qualité du code (semaines 14-15)

LISEZ ATTENTIVEMENT CE DOCUMENT. IL EST LONG, MAIS IL DÉCRIT EN DÉTAILS LA LOGIQUE QUE VOTRE APPLICATION DEVRA IMPLÉMENTER.

VOUS VOUS RÉFÉREREZ À CE DOCUMENT TOUT AU LONG DE LA CONCEPTION ET DE LA RÉALISATION DU PROJET.

Table des matières

Phase 1: Cahier de charges, analyse et séparation générale des tâches.....	1
Rappel.....	1
Consignes de la phase #1.....	3
Analyse - description.....	4
Cahier de charges.....	5
Contexte du projet.....	5
Concepts et entités.....	6
Clients.....	6
Entrepôts.....	6
Produits.....	6
Fournisseurs.....	7
Ordre d'expédition (Shipment orders).....	7
Expéditions (shipments).....	8
Ordres de restockage (purchase order).....	8
Adresses enregistrées.....	8
Utilisateurs.....	9
Roles.....	9
Requis de l'application et description des processus.....	10
Démarrage de l'application et connexion au système.....	10
Gestion des clients.....	10
Nouveaux produits et fournisseurs.....	10
Réception d'une commande et création des ordres d'expédition.....	11
Complétion des ordres d'expédition (Shipping orders).....	11
Complétion des ordres de restockage (Purchase orders).....	12
Administrateurs.....	12
À remplir.....	13
Séparation des tâches.....	13
Classes du domaine.....	14
Liste des cas d'utilisation.....	15
Exemples pour la section « À remplir ».....	18
Séparation des tâches – exemple.....	18
Classes du domaine – exemple.....	19
Liste des cas d'utilisation - exemple.....	21
Section 1 : Gestion des patients.....	21
Section 2 : Gestion des Médecins.....	23
Section 3 : Gestion des rendez-vous.....	25

Consignes de la phase #1

Vous avez jusqu'à la semaine #7 du cours pour compléter la phase d'analyse (phase #1) (compréhension du projet et conception logique de l'application). Lisez attentivement la section Cahier de charges du présent document. Analysez la description de l'application et de ses requis pour en définir les **cas d'utilisation**. Discutez en groupe afin de mieux cerner ceux-ci et/ou de partager vos idées de cas d'utilisation.

Remplissez la section À remplir du présent document. **Une copie dudit document avec cette section complétée doit être remise à l'enseignant par le/la coordonnateur(trice) de l'équipe.** Vous devez :

- **Vous séparer globalement le travail de réalisation entre les membres de l'équipe.** Une grande part du projet est composé de 'stacks' centrées sur les entités du domaine. Lors de la phase de réalisation du projet, chaque classe entité du domaine aura aussi une classe d'accès au données (DAO), une classe métier (Service) et une classe de présentation (View). Chaque équipier devra, au cours de la phase de réalisation du projet, réaliser individuellement deux de ces 'stacks' au complet. Les autres parties communes nécessaires au fonctionnement de l'application devraient être réalisées conjointement.
- **Lister les classes qui formeront les entités du domaine de votre application avec leurs attributs/propriétés ainsi que leurs méthodes** (seulement pour les entités du domaine, pas pour les classes d'accès aux données, les services ou les classes de présentation).
- **Lister les cas d'utilisation que vous prévoyez pour l'application**; en gros toutes les fonctionnalités de base / interactions qu'un utilisateur peut faire avec l'application, sous forme de points

Il n'y a pas de points attribués à la quantité ou à l'exactitude des cas d'utilisation, le but de cette phase est de vous faire réfléchir à ce que votre application va devoir faire. **Il y a, cependant, un malus pouvant aller jusqu'à 20% sur la note du projet sera attribué aux équipes ne remettant pas de liste de cas d'utilisation ou qui en remettent une grossièrement incomplète.** Selon les cas/situations qui me seraient communiquées, ce malus peut être appliqué individuellement. J'ose espérer qu'il n'y aura pas d'équipe avec assez de problèmes internes pour en arriver là, mais je reste flexible à toutes les possibilités.

Dans tous les cas, avant d'arriver à la phase de réalisation du projet, je vous remettrai un document d'analyse complet (c'est-à-dire, comment j'imagine, personnellement, que l'application devrait fonctionner). Donc, faites de votre mieux, mais sachez que si vous en arrachez dans l'analyse et la conception, ou que vous n'êtes vraiment pas sûrs de ce que vous faites, ça ne vous pénalisera pas; **au pire, vous pourrez toujours implémenter mon modèle.**

Analyse - description

La phase d'analyse du projet sert à cerner les besoins généraux en termes de fonctionnalités de votre application, ce qu'elle devra accomplir; elle consiste en **la lecture attentive du cahier de charges de l'application**, en **la conceptualisation des classes entités du domaine de l'application** et en **la rédaction d'une liste des cas d'utilisations prévus de l'application**.

Les cas d'utilisation sont des descriptions textuelles abrégées de scénarios ou groupes de scénarios qui décrivent comment un acteur (utilisateur, administrateur ou système externe) utilise une application pour atteindre un but, incluant le résultat (réussites et/ou échecs). Pour le cadre de ce cours, ne vous est demandé que de donner un nom au cas d'utilisation (généralement, celui-ci commence par un verbe) et de fournir un scénario par étapes générales des interactions entre le/les acteurs et l'application pour atteindre le but.

Il est important de noter que **ces scénarios NE SONT PAS FINAUX**. Ils peuvent être modifiés ultérieurement au cours du processus de développement.

Exemples (juste le nom du cas d'utilisation et qui sont hors-sujet par rapport au cahier de charges) :

- Créer un nouveau patient
- Rechercher un patient par identifiant
- Rechercher un patient par nom
- Modifier un patient
- Supprimer un patient
- Enregistrer un rendez-vous pour un patient
- Modifier un rendez-vous pour un patient
- ...

INDICE : Normalement, il y aura au moins quatre cas d'utilisation pour chaque classe entité du domaine de l'application, ne serait-ce que pour en faire la gestion de base (opérations CRUD). Il devrait y avoir aussi quelques cas d'utilisation supplémentaires pour les cas de fonctionnalités plus complexes de votre application.

Cahier de charges

Contexte du projet

La compagnie (**fictive, NDLR**) *WeStoreYouSell Inc.* (abrégé *WSYS*) œuvre dans le domaine de la logistique d'entreprise et offre des services de gestion d'entrepôt pour des compagnies de vente commerciale clientes qui ne disposent pas d'installations d'entreposage. La compagnie possède plusieurs entrepôts dans différentes villes canadiennes qui desservent des clients locaux.

WSYS offre des services de stockage, de maintien des stocks et de réalisation de commandes; c'est à dire que, la compagnie maintient dans ses entrepôts des stocks de produits pour ses clients (entreprises de vente). Sur réception de commandes passées par les clients (consommateurs) de ses clients (entreprises de vente) par les employés de bureau de *WSYS*, les employés d'entrepôt de *WSYS* ramassent les produits des commandes, les emballent et les envoient pour livraison au consommateur.

La compagnie veut se doter d'une application informatique interne pour la gestion de ses opérations par ses employés (utilisateurs). Les employés de celle-ci sont divisés en trois groupes ayant un rôle différent :

- les employés de bureau : ils gèrent les clients, les produits des clients, les fournisseurs des clients, la création des ordres d'expédition et la création d'ordres de restockage manuels;
- les employés d'entrepôt : ils gèrent la réalisation des ordres d'expédition, c'est-à-dire ramassent et emballent les produits d'un ordre d'expédition (*shipping order*) dans un colis, requièrent le ramassage et l'expédition du colis par des services de livraison externes et les complètent les ordres d'expéditions une fois leur colis ramassé par le service de livraison externe. Ils gèrent aussi la complétion des ordres de restockage lorsque leur entrepôt reçoit une livraison de produits; et
- les administrateurs : ils ont accès à toutes les fonctionnalités du système, notamment de gérer les utilisateurs et les rôles du système.

(**NDLR** : la liste des éléments spécifiés dans la description de chaque concept n'est pas exhaustive, vous pourrez en ajouter à votre guise ou si le fonctionnement de l'application le requiert)

Concepts et entités

Clients

Les clients de la compagnie sont des compagnies de vente sans installations d'entreposage physiques, elles sont représentées par :

- un identifiant interne unique (automatique),
- un nom de compagnie,
- un entrepôt attiré,
- un nom d'une personne-contact,
- un prénom d'une personne-contact,
- un courriel d'une personne-contact,
- un téléphone d'une personne-contact,
- une date de création (automatique),
- une date de modification (automatique), et
- une date de suppression (automatique).

Note : Les clients (compagnies de vente) communiquent les commandes de leurs propres clients (consommateurs) aux employés de WSYS par l'entremise de courriels. L'application n'aura pas à gérer cette partie du processus, seulement le travail de l'employé de bureau pour créer des ordres d'expédition pour ces commandes.

Entrepôts

La compagnie WSYS gère plusieurs entrepôts à des adresses différentes à travers le Canada dans lesquels travaillent des employés d'entrepôt. Chaque entrepôt gère individuellement les stocks des produits de ses clients attirés. Ils sont représentés par :

- un identifiant interne unique (automatique),
- un nom d'entrepôt,
- une adresse enregistrée,
- une date de création (automatique),
- une date de modification (automatique), et
- une date de suppression (automatique).

Produits

Les produits des clients de la compagnie stockés dans ses entrepôts sont représentés par :

- un identifiant interne unique (automatique),
- un nom de produit,
- une description du produit,
- un code UPC international,
- un nom de fichier d'image du produit (peut être laissé vide : **Indice - nullable**),
- un client auquel le produit appartient,
- un fournisseur,
- un code produit du fournisseur,
- une quantité en stock,
- une quantité de stock visée,
- un poids en kilogrammes,

- une date de création (automatique),
- une date de modification (automatique), et
- une date de suppression (automatique).

Il est à noter que des clients différents peuvent avoir le même produit physique dans le même entrepôt; ces produits sont cependant enregistrés séparément dans le système.

Fournisseurs

Chaque produit est associé à un fournisseur utilisé pour créer des commandes de restockage. Les fournisseurs sont représentés par :

- un identifiant interne unique (automatique),
- un nom de fournisseur,
- un nom d'une personne-contact,
- un prénom d'une personne-contact,
- un courriel d'une personne-contact,
- un téléphone d'une personne-contact,
- une date de création (automatique),
- une date de modification (automatique), et
- une date de suppression (automatique).

Ordre d'expédition (Shipment orders)

Les ordres d'expédition sont créées par les employés de bureau de la compagnie lorsqu'ils reçoivent les informations des commande des clients. Ils représentent l'opération interne à la compagnie de ramassage, d'emballage et d'envoi par les employés d'entrepôt. Lorsqu'un ordre d'expédition est créé, la quantité en stock des produits est diminuée du nombre inclus dans l'ordre. Ils sont représentés par :

- un identifiant interne unique (automatique),
- un statut (valeurs possibles : new, unassigned, processing, packaged, shipped)
- un client,
- un utilisateur employé de bureau ou administrateur (le créateur de l'ordre d'expédition),
- une liste de liens produits-ordre d'expédition (**indice – table pivot avec 3^e colonne pour la quantité du produit dans l'ordre**),
- une adresse enregistrée (celle du destinataire final),
- un utilisateur employé d'entrepôt attitré (attitré lorsqu'un employé d'entrepôt prend charge de cet ordre d'expédition – **indice : nullable**),
- une date d'expédition (remplie par un employé d'entrepôt lorsque le service d'expédition externe est venu chercher le colis (complète la commande) – **Indice : nullable**),
- une date de création (automatique),
- une date de modification (automatique), et
- une date de suppression (automatique).

Expéditions (shipments)

Les expéditions sont les livraisons par des services externes des colis des ordres d'expédition, générés par les employés d'entrepôt lorsqu'ils ont fini de ramasser et d'emballer les produits de ceux-ci. Elles sont représentées par :

- un identifiant interne unique (automatique),
- un service de livraison (valeurs possibles : Purolator, PostesCanada, FedEx)
- un code de suivi généré automatiquement par le service de livraison,
- un ordre d'expédition attitrée,
- une date de création (automatique),
- une date de modification (automatique), et
- une date de suppression (automatique).

(NDLR : on ne va pas réellement implémenter les services externes (APIs) des services de livraisons du monde réel, c'est bien trop compliqué, le projet initial va vous fournir un moyen de faire générer des faux codes de suivi que vous pourrez utiliser).

Ordres de restockage (purchase order)

Lorsque la quantité en stock d'un produit tombe en bas de 50% de la quantité à maintenir en stock du même produit (par suite de création d'un ordre d'expédition), un ordre de restockage doit être créé automatiquement pour restocker les produits. Ceux-ci sont représenté par :

- un identifiant interne unique (automatique),
- un statut (valeurs possible : new, completed),
- un produit à restocker,
- un entrepôt (celui où le produit est stocké),
- une quantité à commander,
- une date de complétion de l'ordre de restockage,
- une date de création (automatique),
- une date de modification (automatique), et
- une date de suppression (automatique).

Adresses enregistrées

La compagnie enregistre les adresses civiques de ses clients, des fournisseurs et de ses entrepôts. Ces adresses enregistrées sont représentées par :

- un identifiant interne unique (automatique),
- un type d'adresse (valeurs possibles : entrepot, destinataire),
- un destinataire (nom du destinataire/compagnie/bâtiment...),
- un numéro civique (peut contenir des lettres),
- une rue,
- une ville,
- une province,
- un pays,
- un code postal,
- une date de création (automatique),
- une date de modification (automatique), et
- une date de suppression (automatique).

Utilisateurs

Les utilisateurs représentent les employés de la compagnie qui utilisent le système. Ils sont représentés par :

- un identifiant interne unique (automatique),
- un nom d'utilisateur lui aussi unique,
- un *hash* de mot de passe (Note : je vous fournirai le code pour le hashage (encryptage) et la validation des mots de passe, ne vous cassez pas la tête avec ça. Le *hash* est simplement un string ≤ 128 caractères),
- un ou plusieurs rôles attribué,
- un entrepôt (celui où l'utilisateur travaille pour un employé d'entrepôt, vide pour les employés de bureau),
- une date de création (automatique),
- une date de modification (automatique), et
- une date de suppression (automatique).

Rôles

Les rôles représentent les rôles des employés : de bureau, d'entrepôt ou administrateurs. Ils sont représentés par :

- un identifiant interne unique (automatique),
- un nom,
- une description,
- une date de création (automatique),
- une date de modification (automatique), et
- une date de suppression (automatique).

Requis de l'application et description des processus

Cette section décrit de manière générale les processus de la compagnie et le fonctionnement de l'application tel que désiré.

Démarrage de l'application et connexion au système

Lors du démarrage de l'application, le système doit requérir à l'utilisateur de s'authentifier avec son nom d'utilisateur et mot de passe. Une fois authentifié, une fenêtre principale différente selon le rôle de l'utilisateur authentifié devra s'ouvrir. Celles-ci fourniront l'interface principale pour les fonctionnalités requises par les tâches de l'employé.

Les fonctionnalités à implémenter seront soit accessibles directement depuis les menu principaux des utilisateurs soit via des fenêtres modales (*pop-up*) ouvertes contextuellement par des actions des utilisateurs.

Gestion des clients

Les employés de bureau de WSYS ont la tâche de la gestion des clients de la compagnie. Ce sont eux qui créent les nouveaux clients dans le système. La création d'un client implique l'entrée des données de base du client, mais aussi l'assignation d'un entrepôt existant de WSYS dans lequel seront stockés les produits de ce client.

Notes :

- On ne peut pas modifier à quel entrepôt un client est assigné (trop compliqué pour le cadre du cours, il faudrait aussi modifier tous les produits du client)

Nouveaux produits et fournisseurs

Les employés de bureau de WSYS ont la tâche de la gestion des produits des clients et des fournisseurs dans le système. Ce sont eux qui créent les nouveaux produits dans le système.

La création d'un produit implique l'entrée des données de base du produit, mais aussi la sélection d'un client existant auquel appartiendra le produit, la sélection d'un fournisseur existant pour le produit OU la création d'un nouveau fournisseur, ainsi que la sélection, optionnellement, d'un fichier d'image pour le produit.

Notes :

- On ne peut pas modifier à quel client un produit appartient.
- Le fichier d'image sélectionné lors de la création d'un produit devra être copié dans un dossier spécifique du projet pour être accessible par la suite.

Les employés de bureau peuvent aussi créer manuellement des ordres de restockage pour des produits (qui normalement sont créés automatiquement quand la quantité en stock d'un produit baisse en dessous d'une certaine limite).

Réception d'une commande et création des ordres d'expédition

Les clients (compagnies de ventes) de WSYS communiquent avec les employés de bureau de celle-ci pour leur faire parvenir les commandes de leurs propres clients (consommateurs) qui contiennent la liste des produits de la commande et leur quantité, ainsi que l'adresse du destinataire final (consommateur) auquel la commande devra être livrée.

Lorsqu'un employé de bureau reçoit une commande de la part d'un des clients de la compagnie, il crée un nouvel ordre d'expédition. L'utilisateur créateur de l'ordre d'expédition ainsi que le statut « new » sont automatiquement assignés à l'ordre d'expédition créé; de plus, la création d'un ordre d'expédition implique la sélection (recherche & sélection) ou la création d'une adresse pour le destinataire final (consommateur).

Une fois l'ordre d'expédition créé, l'employé de bureau procède à la sélection puis l'ajout de produits et de leur quantité pour l'ordre d'expédition. L'ajout de produits et de leur quantité à un ordre d'expédition modifie la quantité en stock de ces produits. Selon la quantité restante en stock des produits, des ordres de restockages sont créés automatiquement pour les produits dont la quantité restante est inférieure à 50% de leur quantité à maintenir.

Lorsque tous les produits et quantités sont ajoutés à l'ordre d'expédition, l'employé de bureau la confirme, ce qui fait passer son statut à « unassigned »; et la tâche de l'employé de bureau pour la création d'ordres d'expédition est terminée.

Complétion des ordres d'expédition (Shipping orders)

Les ordres d'expédition dont le statut est « unassigned » sont listés dans une section du menu principal des employés d'entrepôt travaillant dans le même entrepôt que celui associé au client de l'ordre d'expédition. Les employés d'entrepôts peuvent alors s'attirer ces ordres (dont le statut change à « processing »). Les ordres d'expédition attirés à un employé d'entrepôt mais dont le ramassage n'est pas complété sont listés dans une seconde section du menu principal de l'employé d'entrepôt concerné (et seulement de lui).

L'employé d'entrepôt en question va alors ramasser (pick-up) les produits de l'ordre d'expédition dans l'entrepôt et les emballe. Une fois l'emballage complété, il sélectionne un service de livraison et fait générer une expédition (*shipment*) pour l'ordre d'expédition, dont le statut passe alors à « packaged », et continue son travail avec d'autres ordres d'expédition.

Les ordres d'expédition dont le statut est « packaged » sont listés dans une autre section de tous les employés d'entrepôt qui travaillent dans l'entrepôt de ces ordres d'expédition. Lorsque le service de livraison externe vient ramasser un de ces ordres d'expédition, un employé d'entrepôt marque l'ordre d'expédition comme complété, ce qui change son statut à « shipped ».

Complétion des ordres de restockage (Purchase orders)

Les ordres de restockage de produits stockés dans un entrepôt dont le statut est « new » sont listés dans le menu principal des employés d'entrepôt du même entrepôt. lorsque les produits de ces ordres de restockage sont reçus par l'entrepôt, un employé d'entrepôt va les ranger dans l'entrepôt et marquer l'ordre de restockage comme complété ce qui va changer son statut à « completed ». La complétion d'un ordre de restockage modifie la quantité en stock du produit de l'ordre de restockage.

Administrateurs

Les utilisateurs-administrateurs ont accès à un menu principal aux fonctionnalités étendu qui leur permet d'accéder à toutes les fonctionnalités de l'application (gestion de toutes les structures / entités du domaines de l'application), incluant la création d'utilisateurs et de rôles.

La création d'un utilisateur implique la sélection d'un ou de plusieurs rôles existants à associer à l'utilisateur.

À remplir

Séparation des tâches

Dans cette section, vous devez vous séparer les tâches générales (les « stacks ») pour la définition des cas d'utilisation et pour la réalisation future de l'application elle-même en remplissant la table ci-dessous. **Une seule personne doit être assignée à chaque « stack ».** Ajoutez aussi les noms d'utilisateur Github dans la colonne appropriée; ceci sera utilisé pour le partage du projet et pour la correction.

Comme certaines entités sont plus nécessaires que d'autres au fonctionnement – et donc aux tests – de l'application (utilisées pour accéder au programme ou utilisées par d'autres entités), je les ai listé en ordre de priorité : plus une « stack » est haute dans la table, plus sa priorité est haute. Cela vous permet d'assigner les « stacks » les plus prioritaires aux équipiers ayant le plus de temps pour travailler sur le projet ou qui sont plus motivés.

Pour les équipes de 4 où c'est l'enseignant qui agira à titre de 5^e équipier, il est automatiquement assigné aux tâches les plus prioritaires afin de minimiser les délais et d'assurer que les sections primordiales de l'application seront complétées.

Pour les équipes de 5, assignez simplement l'un d'entre vous à la place de l'enseignant (effacez le texte pré-rempli dans la colonne des équipiers assignés).

Tâche ("Stack" Entité – DAO – Service – Vue)	Équipier assigné	Nom d'utilisateur github
Utilisateurs	(professeur si équipe de 4)	
Rôles	(professeur si équipe de 4)	
Clients		
Adresses		
Entrepôts		
Fournisseurs		
Produits		
Expéditions		
Ordres d'expédition		
Ordres de restockage		

Classes du domaine

Vous devez aussi schématiser les classes-entités du domaine de l'application et des relations entre elles (indice : voir la section Concepts et entités du présent document). Utilisez un logiciel de diagrammes pour réaliser ce schéma sous la forme d'un diagramme de classes. Dans un diagramme de classes, les classes sont représentées par des boîtes ayant des sous-sections pour les attributs (champs ou propriétés) et les méthodes que possèdent ces classes. **Note : vous n'avez à schématiser que les classes-entités du domaine; pas les DAOs, services et classes de présentation.**

Vous pouvez soit incorporer dans le document présent une image du schéma ou, si vous utilisez le logiciel que je vous suggère ci-dessous, inclure directement le fichier du diagramme dans une archive (zip) pour la remise de cette phase du projet.

Je vous suggère fortement d'utiliser le logiciel draw.io pour réaliser ce schéma pour les raisons suivantes :

- Il est gratuit.
- Il est disponible directement en ligne à l'adresse app.diagrams.net. Vous pouvez travailler sur des diagrammes en ligne sans télécharger d'application.
- Il permet d'enregistrer des diagrammes directement dans Google Drive, de partager ces diagrammes et de travailler à plusieurs simultanément sur le même diagramme partagé (dans sa version en ligne).
- Il supporte les diagrammes de classes, propose un type de diagramme pré-fait pour ceux-ci (*class diagram*) et offre des composantes UML de diagrammes de classe à simplement cliquer-déposer (*drag-and-drop*) dans le schéma.
- Il est facile à utiliser avec des menus contextuels (clic-droit) sur les éléments du diagramme.
- Il permet d'exporter les diagrammes sous forme d'image JPEG standard et/ou de les enregistrer dans son format propriétaire (.drawio).

Voir la section Classes du domaine – exemple pour un exemple de diagramme de classe réalisé avec *draw.io*.

Liste des cas d'utilisation

Note: Recopiez la table d'informations d'un cas d'utilisation autant de fois que nécessaire.

Cas d'utilisation	
Numéro :	
Nom :	
Préalables :	
Description :	1.

Cas d'utilisation	
Numéro :	
Nom :	
Préalables :	
Description :	1.

Cas d'utilisation	
Numéro :	
Nom :	
Préalables :	
Description :	1.

Cas d'utilisation	
Numéro :	
Nom :	
Préalables :	
Description :	1.

Cas d'utilisation	
Numéro :	

Nom :	
Préalables :	
Description :	1.

Cas d'utilisation	
Numéro :	
Nom :	
Préalables :	
Description :	1.

Cas d'utilisation	
Numéro :	
Nom :	
Préalables :	
Description :	1.

Cas d'utilisation	
Numéro :	
Nom :	
Préalables :	
Description :	1.

Cas d'utilisation	
Numéro :	
Nom :	
Préalables :	
Description :	1.

Cas d'utilisation	
-------------------	--

Numéro :	
Nom :	
Préalables :	
Description :	1.

Exemples pour la section « À remplir »

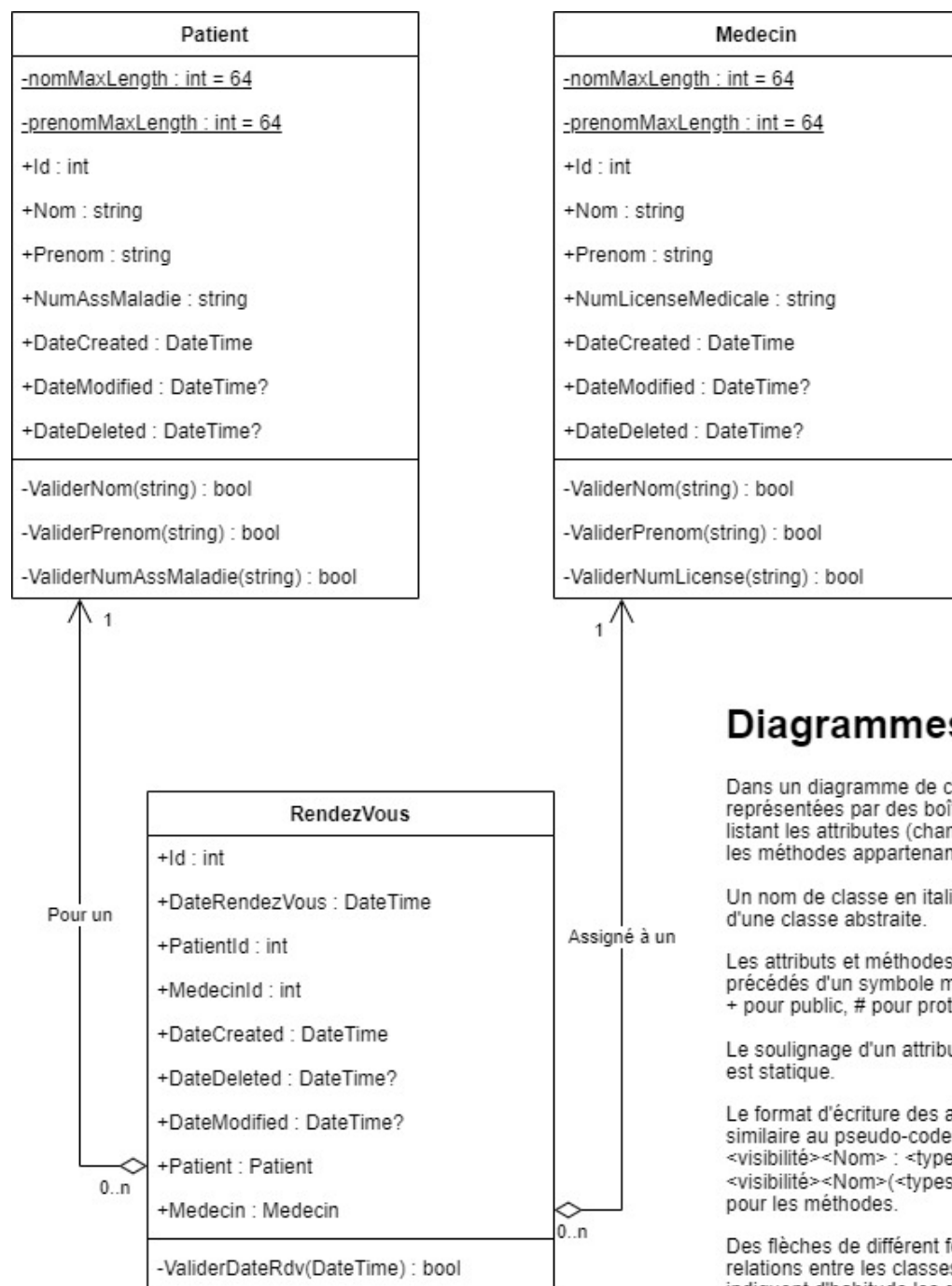
Séparation des tâches – exemple

Tâche (“Stack” Entité – DAO – Service – Vue)	Équipier assigné	Nom d'utilisateur github
Utilisateurs	Mr. Brown ¹	quentinHimself87
Rôles	Mr. Brown	quentinHimself87
Clients	Mr. Pink	angryGuyBuscemi
Adresses	Mr. White	anti_hero
Entrepôts	Mr. Pink	angryGuyBuscemi
Fournisseurs	Mr. Blonde	mad_man_knife
Produits	Mr. Orange	ImNotACop
Expéditions	Mr. Blonde	mad_man_knife
Ordres d'expédition	Mr. Orange	ImNotACop
Ordres de restockage	Mr. White	anti_hero

¹ Ces noms sont une référence au film « Reservoir Dogs » de Quentin Tarantino.

Classes du domaine – exemple

Pour la section d'exemple des classes du domaine et de la liste de cas d'utilisation, j'utiliserai une micro-application minimaliste qui représente un système de gestion des rendez-vous pour des patients et des médecins.



Diagrammes de classes

Dans un diagramme de classes, les classes sont représentées par des boîtes contenant des sections listant les attributs (champs et propriétés) ainsi que les méthodes appartenant à la classe.

Un nom de classe en italique indique qu'il s'agit d'une classe abstraite.

Les attributs et méthodes de la classe listés sont précédés d'un symbole marquant leur accessibilité: + pour public, # pour protected et - pour private.

Le soulignage d'un attribut ou méthode indique qu'il est statique.

Le format d'écriture des attributs et méthodes est similaire au pseudo-code:
 <visibilité><Nom> : <type> pour les attributs, et
 <visibilité><Nom>(<typesParams>) : <typeRetour>
 pour les méthodes.

Des flèches de différents formats représentent les relations entre les classes. Ces flèches de relation indiquent d'habitude les quantités et on souvient une étiquette expliquant la nature de la relation. Par exemple dans ce diagramme: 0 ou plusieurs RendezVous sont assignés à un Medecin.

Liste des cas d'utilisation - exemple

La micro-application servant d'exemple ici sera aussi incluse à titre d'exemple et de référence, en code fonctionnel, dans le projet initial à partir duquel vous développerez l'application de WSYS.

Section 1 : Gestion des patients

Cas d'utilisation	
Numéro :	1.1
Nom :	Créer un patient
Préalables :	<ul style="list-style-type: none"> L'utilisateur de l'application est authentifié et autorisé (Note : pas dans l'exemple)
Description :	<ol style="list-style-type: none"> L'utilisateur choisit l'option de création de nouveau patient dans la fenêtre principale. Le système ouvre la fenêtre de gestion d'un patient en mode création. L'utilisateur entre les données représentant le patient dans les champs à cet effet et : <ol style="list-style-type: none"> 3.a Cliquez sur le bouton créer. <ol style="list-style-type: none"> Le système insère le nouveau patient dans la base de données Le système ferme puis ré-ouvre la fenêtre de gestion d'un patient en mode affichage pour le patient créé. 3.b Cliquez sur le bouton annuler. <ol style="list-style-type: none"> Le système ferme la fenêtre de gestion d'un patient.

Cas d'utilisation	
Numéro :	1.2
Nom :	Voir les informations d'un patient
Préalables :	<ul style="list-style-type: none"> L'utilisateur de l'application est authentifié et autorisé (Note : pas dans l'exemple)
Description :	<ol style="list-style-type: none"> L'utilisateur entre des données (nom ou prénom ou numéro d'assurance maladie) de recherche d'un patient dans un champ prévu à cet effet dans la fenêtre principale. Le système affiche une liste des patients concordant à la recherche. L'utilisateur sélectionne un patient dans la liste affichée et clique sur le bouton approprié pour la visualisation des informations du

	<p>patient.</p> <p>4. Le système ouvre la fenêtre de gestion d'un patient en mode affichage pour le patient sélectionné.</p>
--	--

Cas d'utilisation	
Numéro :	1.3
Nom :	<i>Modifier un patient</i>
Préalables :	<ul style="list-style-type: none"> L'utilisateur de l'application est authentifié et autorisé (Note : pas dans l'exemple)
Description :	<ol style="list-style-type: none"> L'utilisateur entre des données (nom, prénom ou numéro d'assurance maladie) de recherche d'un patient dans un champ prévu à cet effet dans la fenêtre principale. Le système affiche une liste des patients concordant à la recherche. L'utilisateur sélectionne un patient dans la liste affichée et clique sur le bouton approprié pour la modification des informations du patient. Le système ouvre la fenêtre de gestion d'un patient en mode modification pour le patient sélectionné. L'utilisateur apporte des changements aux informations du patient dans les champs prévus à cet effet et : <ol style="list-style-type: none"> 5.a Cliquez sur le bouton sauvegarder. <ol style="list-style-type: none"> 1. Le système met à jour le patient dans la base de données et ré-ouvre la fenêtre de gestion d'un patient en mode affichage pour le patient modifié. 5.b Cliquez sur le bouton annuler. <ol style="list-style-type: none"> 1. Le système ferme la fenêtre de gestion d'un patient.

Cas d'utilisation	
Numéro :	1.4
Nom :	<i>Supprimer un patient</i>
Préalables :	<ul style="list-style-type: none"> L'utilisateur de l'application est authentifié et autorisé (Note : pas dans l'exemple)
Description :	<ol style="list-style-type: none"> L'utilisateur entre des données (nom ou prénom ou numéro d'assurance maladie) de recherche d'un patient dans un champ prévu à cet effet dans la fenêtre principale. Le système affiche une liste des patients concordant à la recherche.

	<ol style="list-style-type: none"> 3. L'utilisateur sélectionne un patient dans la liste affichée et clique sur le bouton approprié pour la suppression des informations du patient. 4. Le système ouvre la fenêtre de gestion d'un patient en mode suppression pour le patient sélectionné. 5. L'utilisateur vérifie les informations et soit : <ol style="list-style-type: none"> 5.a Appuie sur le bouton supprimer. <ol style="list-style-type: none"> 1. Le système requière une confirmation à l'utilisateur 2. L'utilisateur confirme la suppression. <ol style="list-style-type: none"> 2.a Le système supprime le patient dans la base de données. 2.b Le système ferme la fenêtre de gestion d'un patient. 2.c Le système retire le patient la liste de résultats de recherche de patient de la fenêtre principale. 3. L'utilisateur annule la suppression. 5.b L'utilisateur appuie sur le bouton annuler <ol style="list-style-type: none"> 1. Le système ferme la fenêtre de gestion d'un patient.
--	---

Section 2 : Gestion des Médecins

Cas d'utilisation	
Numéro :	2.1
Nom :	Créer un médecin
Préalables :	<ul style="list-style-type: none"> • L'utilisateur de l'application est authentifié et autorisé (Note : pas dans l'exemple)
Description :	<ol style="list-style-type: none"> 1. L'utilisateur choisit l'option de création de nouveau médecin dans la fenêtre principale. 2. Le système ouvre la fenêtre de gestion d'un médecin en mode création. 3. L'utilisateur entre les données représentant le médecin dans les champs à cet effet et : <ol style="list-style-type: none"> 3.a Clique sur le bouton créer. <ol style="list-style-type: none"> 1. Le système insère le nouveau médecin dans la base de données 2. Le système ferme puis ré-ouvre la fenêtre de gestion d'un médecin en mode affichage pour le patient créé. 3.b Clique sur le bouton annuler. <ol style="list-style-type: none"> 1. Le système ferme la fenêtre de gestion d'un médecin.

Cas d'utilisation	
Numéro :	2.2
Nom :	<i>Voir les informations d'un médecin</i>
Préalables :	<ul style="list-style-type: none"> • <i>L'utilisateur de l'application est authentifié et autorisé (Note : pas dans l'exemple)</i>
Description :	<ol style="list-style-type: none"> 1. L'utilisateur entre des données (nom, prénom ou numéro de licence médicale) de recherche d'un médecin dans un champ prévu à cet effet dans la fenêtre principale. 2. Le système affiche une liste des médecins concordant à la recherche. 3. L'utilisateur sélectionne un médecin dans la liste affichée et clique sur le bouton approprié pour la visualisation des informations du médecin. 4. Le système ouvre la fenêtre de gestion d'un médecin en mode affichage pour le médecin sélectionné.

Cas d'utilisation	
Numéro :	2.3
Nom :	<i>Modifier un médecin</i>
Préalables :	<ul style="list-style-type: none"> • <i>L'utilisateur de l'application est authentifié et autorisé (Note : pas dans l'exemple)</i>
Description :	<ol style="list-style-type: none"> 1. L'utilisateur entre des données (nom ou prénom ou numéro de licence médicale) de recherche d'un médecin dans un champ prévu à cet effet dans la fenêtre principale. 2. Le système affiche une liste des médecins concordant à la recherche. 3. L'utilisateur sélectionne un médecin dans la liste affichée et clique sur le bouton approprié pour la modification des informations du médecin. 4. Le système ouvre la fenêtre de gestion d'un médecin en mode modification pour le médecin sélectionné. 5. L'utilisateur apporte des changements aux informations du médecin dans les champs prévus à cet effet et : <ol style="list-style-type: none"> 5.a Cliquez sur le bouton sauvegarder. <ol style="list-style-type: none"> 1. Le système met à jour le médecin dans la base de données et ré-ouvre la fenêtre de gestion d'un médecin en mode affichage pour le médecin modifié. 5.b Cliquez sur le bouton annuler. <ol style="list-style-type: none"> 1. Le système ferme la fenêtre de gestion d'un médecin.

Cas d'utilisation	
Numéro :	1.4
Nom :	<i>Supprimer un médecin</i>
Préalables :	<ul style="list-style-type: none"> L'utilisateur de l'application est authentifié et autorisé (Note : pas dans l'exemple)
Description :	<ol style="list-style-type: none"> L'utilisateur entre des données (nom ou prénom ou numéro de licence médicale) de recherche d'un médecin dans un champ prévu à cet effet dans la fenêtre principale. Le système affiche une liste des médecins concordant à la recherche. L'utilisateur sélectionne un médecin dans la liste affichée et clique sur le bouton approprié pour la suppression des informations du médecin. Le système ouvre la fenêtre de gestion d'un médecin en mode suppression pour le médecin sélectionné. L'utilisateur vérifie les informations et soit : <ol style="list-style-type: none"> Appuie sur le bouton supprimer. <ol style="list-style-type: none"> Le système requière une confirmation à l'utilisateur L'utilisateur confirme la suppression. <ol style="list-style-type: none"> Le système supprime le médecin dans la base de données. Le système ferme la fenêtre de gestion d'un médecin. Le système retire le médecin la liste de résultats de recherche de médecin de la fenêtre principale. L'utilisateur annule la suppression. L'utilisateur appuie sur le bouton annuler <ol style="list-style-type: none"> Le système ferme la fenêtre de gestion d'un médecin.

Section 3 : Gestion des rendez-vous

Cas d'utilisation	
Numéro :	2.1
Nom :	<i>Créer un rendez-vous pour un patient</i>
Préalables :	<ul style="list-style-type: none"> L'utilisateur de l'application est authentifié et autorisé (Note : pas dans l'exemple)
Description :	<ol style="list-style-type: none"> L'utilisateur entre des données (nom ou prénom ou numéro d'assurance maladie) de recherche d'un patient dans un champ prévu à cet effet dans la fenêtre principale.

	<ol style="list-style-type: none"> 2. Le système affiche une liste des patients concordant à la recherche. 3. L'utilisateur sélectionne un patient dans la liste affichée et clique sur le bouton approprié pour la création d'un rendez-vous. 4. Le système ouvre la fenêtre de gestion d'un rendez-vous en mode création pour le patient sélectionné. 5. L'utilisateur sélectionne un médecin et une date pour le rendez-vous, puis: <ol style="list-style-type: none"> 5.a Cliquez sur le bouton créer. <ol style="list-style-type: none"> 1. Le système insère le nouveau rendez-vous dans la base de données. 2. Le système ferme puis ré-ouvre la fenêtre de gestion d'un rendez-vous en mode affichage pour le rendez-vous créé. 5.b Cliquez sur le bouton annuler. <ol style="list-style-type: none"> 1. Le système ferme la fenêtre de gestion d'un rendez-vous.
--	---

Cas d'utilisation	
Numéro :	2.2
Nom :	<i>Voir les informations d'un rendez-vous pour un patient</i>
Préalables :	<ul style="list-style-type: none"> • <i>L'utilisateur de l'application est authentifié et autorisé (Note : pas dans l'exemple)</i>
Description :	<ol style="list-style-type: none"> 1. L'utilisateur entre des données (nom, prénom ou numéro d'assurance maladie) de recherche d'un patient dans un champ prévu à cet effet dans la fenêtre principale. 2. Le système affiche une liste des patients concordant à la recherche. 3. L'utilisateur sélectionne un patient dans la liste affichée et clique sur le bouton approprié pour la visualisation des informations du patient. 4. Le système ouvre la fenêtre de gestion d'un patient en mode affichage pour le patient sélectionné. 5. L'utilisateur sélectionne un rendez-vous dans la liste des rendez-vous à venir du patient affichés dans la fenêtre de gestion d'un patient. Puis clique sur le bouton approprié pour voir les détails du rendez-vous. 6. Le système ouvre la fenêtre de gestion d'un rendez-vous en mode affichage pour le rendez-vous sélectionné.

Cas d'utilisation	
Numéro :	2.3

Nom :	<i>Modifier un rendez-vous</i>
Préalables :	<ul style="list-style-type: none"> • <i>L'utilisateur de l'application est authentifié et autorisé (Note : pas dans l'exemple)</i>
Description :	<ol style="list-style-type: none"> 1. L'utilisateur entre des données (nom, prénom ou numéro d'assurance maladie) de recherche d'un patient dans un champ prévu à cet effet dans la fenêtre principale. 2. Le système affiche une liste des patients concordant à la recherche. 3. L'utilisateur sélectionne un patient dans la liste affichée et clique sur le bouton approprié pour la visualisation des informations du patient. 4. Le système ouvre la fenêtre de gestion d'un patient en mode affichage pour le patient sélectionné. 5. L'utilisateur sélectionne un rendez-vous dans la liste des rendez-vous à venir du patient affichés dans la fenêtre de gestion d'un patient. Puis clique sur le bouton approprié pour modifier le rendez-vous. 6. Le système ouvre la fenêtre de gestion d'un rendez-vous en mode modification pour le rendez-vous sélectionné. 7. L'utilisateur apporte des changements aux informations du rendez-vous dans les champs prévus à cet effet et : <ol style="list-style-type: none"> 7.a Cliquez sur le bouton sauvegarder. <ol style="list-style-type: none"> 1. Le système met à jour le rendez-vous dans la base de données et ré-ouvre la fenêtre de gestion d'un rendez-vous en mode affichage pour le rendez-vous modifié. 7.b Cliquez sur le bouton annuler. <ol style="list-style-type: none"> 1. Le système ferme la fenêtre de gestion d'un rendez-vous.

Cas d'utilisation	
Numéro :	<i>1.4</i>
Nom :	<i>Supprimer un rendez-vous</i>
Préalables :	<ul style="list-style-type: none"> • <i>L'utilisateur de l'application est authentifié et autorisé (Note : pas dans l'exemple)</i>
Description :	<ol style="list-style-type: none"> 1. L'utilisateur entre des données (nom, prénom ou numéro de licence médicale) de recherche d'un patient dans un champ prévu à cet effet dans la fenêtre principale. 2. Le système affiche une liste des patients concordant à la recherche. 3. L'utilisateur sélectionne un patient dans la liste affichée et clique sur le bouton approprié pour la visualisation des informations du

	<p>patient.</p> <ol style="list-style-type: none">4. Le système ouvre la fenêtre de gestion d'un patient en mode affichage pour le patient sélectionné.5. L'utilisateur sélectionne un rendez-vous dans la liste des rendez-vous à venir du patient affichés dans la fenêtre de gestion d'un patient. Puis clique sur le bouton approprié pour supprimer le rendez-vous.6. Le système ouvre la fenêtre de gestion d'un rendez-vous en mode suppression pour le rendez-vous sélectionné.7. L'utilisateur vérifie les informations et soit :<ol style="list-style-type: none">7.a Appuie sur le bouton supprimer.<ol style="list-style-type: none">1. Le système requière une confirmation à l'utilisateur2. L'utilisateur confirme la suppression.<ol style="list-style-type: none">2.a Le système supprime le rendez-vous dans la base de données.2.b Le système ferme la fenêtre de gestion d'un rendez-vous.2.c Le système rafraîchit la liste de rendez-vous à venir dans la fenêtre de gestion d'un patient.3. L'utilisateur annule la suppression.7.b L'utilisateur appuie sur le bouton annuler<ol style="list-style-type: none">1. Le système ferme la fenêtre de gestion d'un médecin.
--	--