IPST-CNAM
Architecture Logicielles
NSY 205
Mercredi 23 Février 2022

Durée : **2 h 30**
Enseignant : LAFORGUE Jacques

1ère Session NSY 205

1ère PARTIE – SANS DOCUMENT (durée: 1h15)
CORRECTION

1. QCM (35 points)

Mode d'emploi :

Ce sujet est un QCM dont les questions sont de 3 natures :

- **les questions à 2 propositions**: dans ce cas une seule des 2 propositions est bonne.
  - +1 pour la réponse bonne
  - –1 pour la réponse fausse
- **les questions à 3 propositions** dont 1 seule proposition est bonne
  - + 1 pour la réponse bonne
  - –½ pour chaque réponse fausse
- **les questions à 3 propositions** dont 1 seule proposition est fausse
  - + ½ pour chaque réponse bonne
  - –1 pour la réponse fausse

Il s'agit de faire une croix dans les cases de droite en face des propositions.

On peut remarquer que cocher toutes les propositions d'une question revient à ne rien cocher du tout (égal à 0).

Si vous devez raturer une croix, faites-le correctement afin qu'il n'y ait aucune ambiguïté.

N'oubliez pas d'inscrire en en-tête du QCM, votre nom et prénom.

Vous avez droit à **4 points** négatifs sans pénalité.

NOM:	PRENOM:

Une architecture logicielle est la représentation d'un programme sous la forme d'un diagramme de classe UML qui décrit les relations entre les composants (les classes) du programme.		Q 1.
1	OUI	
2	NON	X

Dans la démarche d'architecture d'un Système d'Information, l'Architecture Technique est l'implémentation de la Configuration Architecturale dans une ou plusieurs technologies données.		Q 2.
1	OUI	X
2	NON	

Une architecture logicielle est une description des composants d'un Système d'Information, en des termes fonctionnels, logiques, techniques et physiques permettant de maitriser et faire comprendre la réalisation du Système d'Information.		Q 3.
1	OUI	X
2	NON	

L'élément de base d'une architecture logiciel est le composant. Il est indispensable d'identifier **au plus tôt** dans quelle technologie sera implémenté le composant		Q 4.
1	OUI	
2	NON	X

Une **Configuration Architecturale** est composée de 3 parties : l' **Architecture Applicative**, l' **Architecture Technique**, et l' **Architecture Physique**.		Q 5.
1	OUI	
2	NON	X

L'Architecture Fonctionnelle contient la description des exigences "fonctionnelles" et "non-fonctionnelles" identifiées pendant la conception de l'architecture d'un Système d'Information.		Q 6.
1	OUI	X
2	NON	

L' **Architecture Dynamique** est l'architecture logicielle du socle système sur lequel s'appuient tous les composants de l'architecture logicielle		Q 7.
1	OUI	
2	NON	X

Dans la démarche d'architecture d'un Système d'Information, l'Architecture Dynamique :		Q 8.
1	définit sur quoi, et où, s'exécutent les composants de son architecture.	
2	définit le comportement dynamique interne de chacun des composants de son architecture.	X
3	définit la coopération des composants entre eux au sein de son architecture.	X

Quand cela est possible, dans une démarche d'architecture, on essaye de privilégier un couplage fort entre les composants.		Q 9.
1	OUI	
2	NON	X

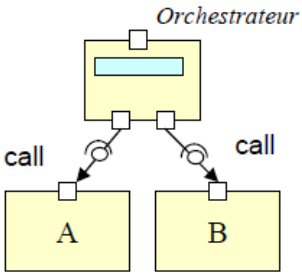
Dans l'Architecture Technique, la conception d'un ORM (Object Relationnal Mapping) permet de réaliser un mapping de programmation efficace entre les données du SI et la Base de Données utilisée.		Q 10.
1	OUI	X
2	NON	

Dans une architecture logicielle, un composant est :		Q 11.
1	une unité de composition logicielle, exposant des interfaces bien spécifiées	X
2	une unité de composition logicielle, susceptible d'être déployée de manière indépendante	X
3	une unité de composition logicielle spécifique qui ne peut plus se décomposer en d'autres unités de composition logicielle	

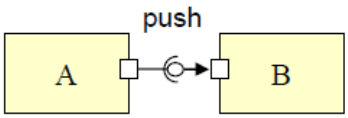
Plusieurs interfaces peuvent être attachées à un même port		Q 12.
1	OUI	X
2	NON	

Dans la démarche d'architecture, un connecteur entre deux composants définit toujours un lien distant de communication de machine à machine.		Q 13.
1	OUI	
2	NON	X

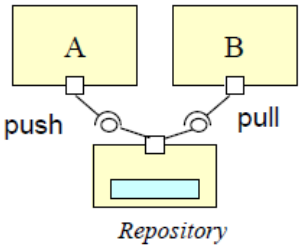
Soit les 3 modèles de coopération de composants suivants:



Modèle 1



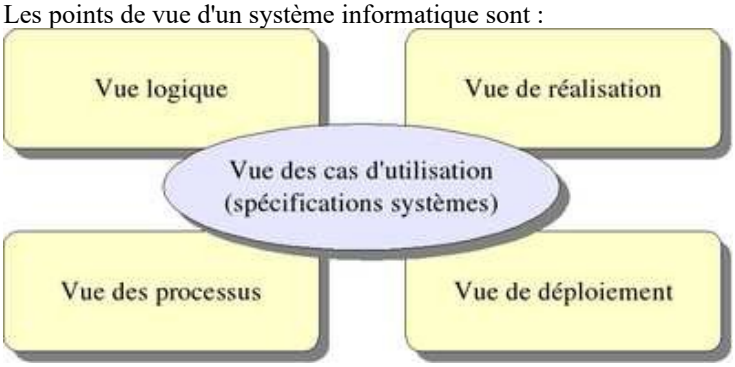
Modèle 2



Modèle 3

Quel est le mode de contrôle de chacun ces modèles :

		Q 14.
1	Modèle 1 = contrôle orienté procédure Modèle 2 = contrôle orienté données Modèle 3 = contrôle orienté évènements	
2	Modèle 1 = contrôle orienté procédure Modèle 2 = contrôle orienté évènements Modèle 3 = contrôle orienté données	X
3	Modèle 1 = contrôle orienté évènements Modèle 2 = contrôle orienté données Modèle 3 = contrôle orienté procédure	

Les points de vue d'un système informatique sont : 		Q 15.
1	La vue de réalisation permet de visualiser l'organisation des constituants dans l'environnement de développement	X
2	La vue des processus décrit les interactions entre les différents processus, threads (files d'exécution) ou tâches	X
3	La vue logique correspond à l'architecture technique d'un Système d'Information	

Dans une architecture J2EE, en EJB Session Stateful est permet de garder en mémoire un certain état de l'EJB.		Q 16.
1	OUI	X
2	NON	

Il existe différentes approches pour réaliser la persistance des objets dans une base de données. Toutes ces approches nécessitent d'utiliser un progiciel (ou COTS) qui réalise automatiquement le mapping des objets en base de données		Q 17.
1	OUI	
2	NON	X

Dans l'approche ORM, le mapping de l'association UML *..* entre deux classes est toujours réalisé par la création d'une table d'association		Q 18.
1	OUI	X
2	NON	

Dans l'approche ORM, le mapping de l'association UML 1..* entre deux classes est toujours réalisé par la création d'une table d'association		Q 19.
1	OUI	
2	NON	X

Dans un ORM (Object Relationnel Mapping), on veut réaliser le mapping d'une relation 1..* entre deux classes, par exemple Departement et Employe (un département contient plusieurs employés). On peut réaliser ce mapping :		Q 20.
1	sans table de jointure, en mettant la clef primaire de la table EMPLOYE dans la table DEPARTEMENT.	
2	sans table de jointure, en mettant la clef primaire de la table DEPARTEMENT dans la table EMPLOYE.	X
3	avec une table de jointure, en mettant les clefs primaires des tables DEPARTEMENT et EMPLOYE dans la table de jointure.	X

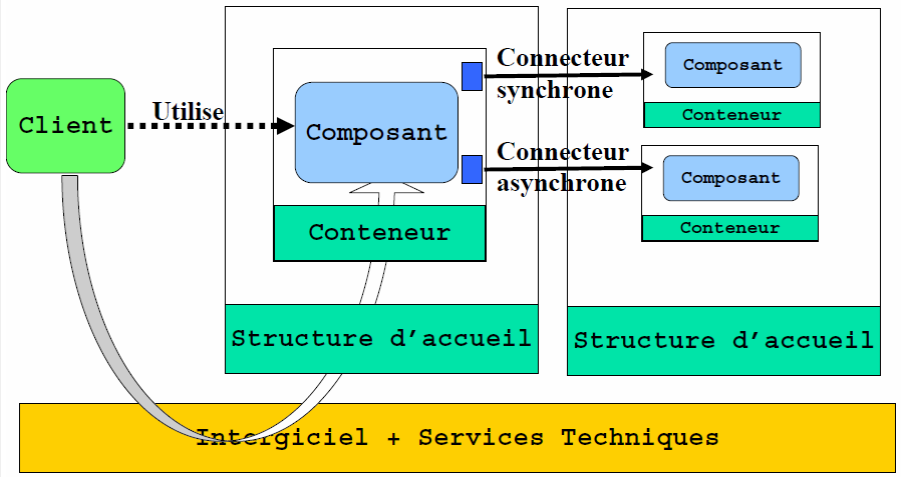
Le mapping ORM de l'héritage de classe peut se faire en créant autant de tables qu'il existe de classes réelles de l'arbre d'héritage		Q 21.
1	OUI	X
2	NON	

Dans une Architecture N-tiers, la couche DAO (Data Access Object) s'intercale entre le tier-IHM et le tier-Métier facilitant pour les clients IHM (Navigateur) l'accès aux données gérées par la couche métier		Q 22.
1	OUI	
2	NON	X

Dans une architecture à base de composant, un des principes de base est d'utiliser les propriétés d'un Framework pour prendre en charge l'exécution des composants de son Système d'Information		Q 23.
1	OUI	X
2	NON	

Un modèle d'architecture J2EE est constitué de différentes couches logicielles, appelées "tiers".		Q 24.
1	Ces couches logicielles sont utilisées de manière horizontale.	X
2	Chacune de ces couches est prise en compte par un "container".	X
3	Ces couches logicielles sont empilées les unes sur les autres et liées entre elles afin de constituer un programme unique appelé "Serveur".	

Un EJB session est un composant d'une architecture J2EE qui est propre à chaque utilisateur Web du serveur d'application		Q 25.
1	OUI	X
2	NON	



Ce schéma présente les principes d'une architecture à base de composant		Q 26.
1	OUI	X
2	NON	

Dans une architecture J2EE, il faut soi-même faire l'instanciation des EJB session Stateful car il faut initialiser les attributs métier de la classe EJB.		Q 27.
1	OUI	
2	NON	X

Un EJB session Stateful est une classe dont les attributs sont tous des Entity. Ainsi les attributs de la classe persistent en base de données, ce qui permet que l'état du EJB Stateful est maintenu en mémoire.		Q 28.
1	OUI	
2	NON	X

Dans une architecture Web Services qui repose sur le protocole SOAP, l'interface de définition d'un service est décrite par :		Q 29.
1	un fichier écrit au standard IDL	
2	un fichier écrit au standard UDDI	
3	un fichier écrit au standard WSDL	**X**

Le protocole SOAP (Simple Object Access Protocol) est un protocole :		Q 30.
1	dont les données échangées peuvent être sérialisées en XML	**X**
2	assurant l'échange d'informations et l'invocation de méthodes distantes (RPC)	**X**
3	assurant l'échange d'informations et l'invocation de méthodes distantes en RMI (Remote Method Invocatoin)	

En REST, la ressource retournée par une requête est "typée", permettant ainsi au client de savoir comment afficher la ressource dans le navigateur		Q 31.
1	OUI	**X**
2	NON	

En REST, le serveur d'application est sans état par rapport aux requêtes traitées. Cet état doit être géré par le client.		Q 32.
1	OUI	**X**
2	NON	

Dans un MOM, les envois de messages sont :		Q 33.
1	toujours synchrone	
2	toujours asynchrone	
3	synchrone ou asynchrone (en fonction du contexte)	**X**

Le principe d'un MOM est :		Q 34.
1	d'utiliser un composant logiciel particulier qui sert d'intermédiaire entre les producteurs et les consommateurs	**X**
2	de lier les producteurs et les consommateurs en faisant une injection de dépendance	
3	d'utiliser le protocole RMI pour réaliser directement l'envoi d'un message du producteur au consommateur	

Dans un MOM, le mode "Queue" de communication par message assure que tous les consommateurs d'un canal d'évènement reçoivent bien le message.		Q 35.
1	OUI	
2	NON	**X**

Fin du QCM

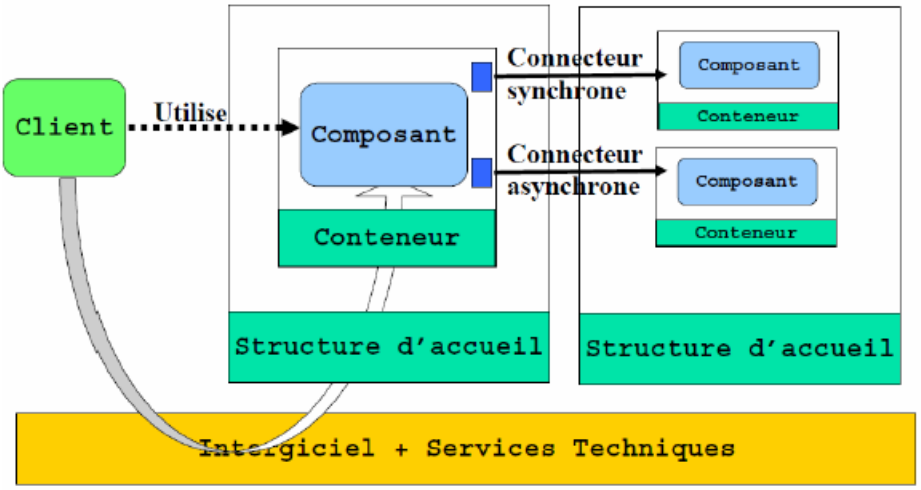
Suite (Tournez la page)

2. Questions libres (15 points)

Chaque question est notée sur 5 points.
*Vous répondez à ces questions sur une **copie vierge double** en mettant bien le numéro de la question, sans oublier votre nom et prénom.*
Vous mettez le QCM dans cette copie vierge double.

QUESTION NUMERO 1

Soit le schéma suivant qui décrit le modèle d'architecture à base de composants (exemple: J2EE) :



Commentez ce schéma tout en mettant en évidence les 2 ou 3 principes forts d'un tel style d'architecture.

Ce schéma représente le modèle d'architecture d'une architecture à base de composant dans lequel les composants dits "métiers" (appelé "Composant" dans le schéma) sont déployés dans des structures d'accueil d'un progiciel, appelé "Serveur d'Application", qui prend en charge l'exécution des composants.

Ce modèle d'architecture facilite la mise en œuvre des SI à base de composant.
Le client utilise les composants métier d'une manière logique en toute transparence de leurs localisations, leur implémentation et leurs dépendances.
Le couplage entre les composants est réalisé par le Serveur d'Application en faisant de l'injection de dépendance entre les composants.

QUESTION NUMERO 2

Nous avons vu dans le cadre des MOM (Middleware Orientés Message), que JMS (Java Messaging System) est une API JEE répondant au principe d'un MOM.

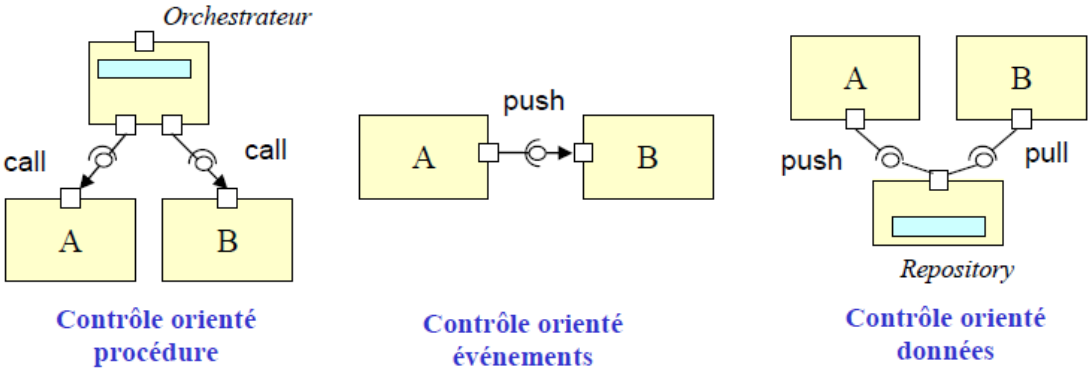
Expliquez quel est le principe général de JMS en utilisant les termes: JMS Client, JMS Provider, JMS Producer, JMS Consumer, JMS Message.

Le principe général de JMS est de créé un composant particulier (unique dans le SI) appelé JMS Provider qui assure la réception et la diffusion des messages appelés JMS Message.

Les clients, appelés JMS Client, se connectent au JMS Provider. Ces clients sont soit des JMS Producer ou des JM Consumer. Les JMS Producer produisent des messages dans un canal d'évènement situé dans le JMS Provider. Les JMS Consumer viennent chercher les messages (mode Queue) ou sont notifiés de ces messages (mode Topic).

QUESTION NUMERO 3

Soit le schéma suivant de représentation de 3 modèles de coopération des composants d'une Architecture Logicielle :



Contrôle orienté procédure **Contrôle orienté événements** **Contrôle orienté données**

Expliquez le comportement de chacun de ces 3 modèles.

Le principe du "Contrôle orienté procédure" est qu'un composant appelé Orchestrateur, fait l'appel aux services du composant A puis du composant B. C'est le principe le plus commun : un composant déroule un algorithme nécessitant l'appel (call) à plusieurs composants de l'architecture.

Le principe du "Contrôle orienté évènements" est que les composants se poussent (push) mutuellement des informations. C'est le principe utilisé dans les architectures RPC quand le push est un call entre A et B via l'interface de B. C'est le principe utilisé dans les architectures MOM quand le push est pris en charge par un Provider de messages.

Le principe du "Contrôle orienté données" est qu'un composant (ici A) dépose une information dans un Repository à un moment donné. Puis à un autre moment, un autre composant (isi B) vient chercher cette information. C'est le principe adopté dans l'utilisation des bases de données.

Fin de la 1ère partie sans document

2ème PARTIE – AVEC DOCUMENT (durée: 1h15)

3. PROBLEME (50 points)

Nous voulons réaliser le prototype d'un Système d'Information permettant de repérer des objets de plus de 3cm qui seraient restés sur une piste d'aviation. Pour cela on utilise des robots (petit rover) munis d'une caméra embarquée, et d'une carte radio goniométrique.
La carte radio goniométrique permet de connaitre au cm près la localisation du robot sur la piste à partir de 3 balises réparties autour de la piste.

Les robots communiquent tous avec un serveur qui centralise les données envoyées par chacun des robots.
Chaque robot fait, en continu, une analyse de ce qu'il observe grâce à un logiciel de reconnaissance des formes. Dans le prototype, ce logiciel est sur le serveur. A terme, il sera embarqué sur le robot pour des raisons de performance. La caméra prend des images à la cadence de 5 images par seconde. La vitesse de déplacement de chacun des robots est configurable.

Les données envoyées par chacun des robots sont : sa position (en continu), et les images.
Les données reçues par chacun des robots sont : les ordres de déplacement.
Chaque robot est piloté par le serveur qui calcule le meilleur parcours en fonction de la position de chacun des robots pour couvrir la zone de recherche.

Une IHM d'administration permet de configurer le nombre et les robots, la topographie de la zone de recherche et la position des balises. Cette IHM permet de visualiser, en temps réel, la position de chacun des robots, sur une carte qui représente la configuration de l'aéroport. Les robots pouvant être utilisés pour un tout autre usage (surveillance par exemple), une commande de joystick connectée à l'ordinateur d'administration, permet de le télécommander.

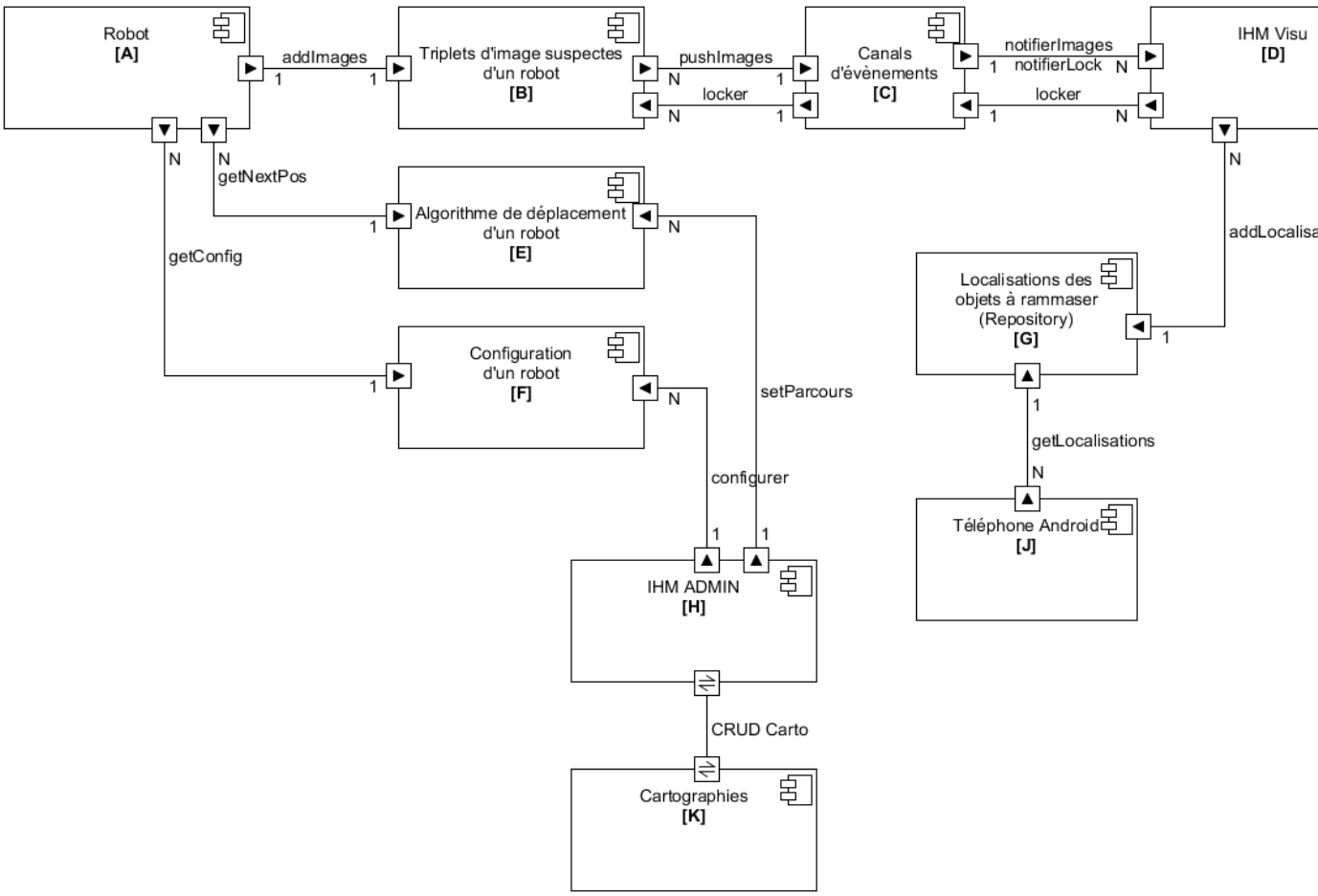
Des IHM de VISU permettent de visualiser les images et la position de chacun des robots.
Dès que le robot a repéré un objet, il s'arrête, prend 3 images sous 3 angles différents qu'il envoie spécifiquement sur le serveur. Puis le robot continue sa route en contournant l'objet.

Le serveur notifie systématiquement toutes les ihms dès qu'un robot a repéré un objet.
Un opérateur peut alors verrouiller le lot d'image afin qu'un autre opérateur sur une autre IHM ne fasse pas le même travaille. L'opérateur peut alors analyser les images et signaler au serveur si l'objet repéré devra être ramassé par un humain.

Une fois que les robots ont ratissé toute la zone. Une Application Android sur un téléphone portable permet de diriger une équipe motorisée sur les objets à ramasser. Cette application récupère du serveur la liste des positions des objets à ramasser et leurs images associées. Cette application permet de valider la prise réelle d'un objet significatif afin d'alimenter une archives des objets trouvés sur lesquels est collé un numéro d'identification (code barre par exemple) .

Le nombre de robot et le nombre d'IHM de VISU sont configurables.

Question 1 :
Faire la **Configuration Architecturale (statique et dynamique)** de ce SI sous la forme d'un schéma d'architecture de composants que vous devez commenter (rôle de chacun des composants, rôle des connecteurs, use-case, ….).
(Pour rendre plus lisible vos commentaires, chaque composant à un nom et chaque connecteur à un nom ou un numéro).



Les rôles des composants, leurs comportements statiques et dynamiques :

[A] Le robot qui récupère sa configuration (getConfig), récupère sa prochaine position (getNextPos), qui ajoute un triplet d'images d'un objet suspect (addImages)
[B] Ce composant reçoit les images suspectes ajoutées par un même robot. Il pousse ses triplets d'image dans un canal d'évènement (pushImages).
[C] Ce composant contient les canaux d'évènement qui notifie les triplets d'image (notifierImages) à toutes les IHM de visualisation. Il notifie aussi comme quoi un triplet d'image a été verrouillé (notifierLock) par une IHM à toutes les IHM de visualisation. Il envoie à tous les composants B (locker) comme quoi un triplet d'image a été verrouillé. Seul le composant [B] concerné par le lock traite cette demande.
[D] Ce composant est l'IHM de visualisation qui affiche toutes les images reçus par notification. Il permet à un opérateur de verrouiller (locker) un triplet d'image afin de les analyser. Si l'analyse est positive alors l'IHM permet à l'opérateur d'ajouter la localisation du triplet d'image suspect (addLocalisation) dans le composant [G]
[E] Ce composant contient l'algorithme qui calcule la liste des positions qu'un robot doit suivre. Cet algorithme a été mis à jour par l'IHM d'administration (setParcours).
[F] Ce composant contient la configuration d'un robot qui a été mise à jour par l'IHM d'administration
[G] Ce composant contient la localisation et les images de tous les objets suspects qui devront être ramassés
[H] Ce composant est l'IHM d'administration qui met à jour la configuration d'un robot (configurer), l'algorithme de déplacement d'un robot (setParcours), de créer, mettre à jour la cartographie des pistes d'aviation (CRUD Carto) afin de déterminer la répartition des robots et donc leurs parcours.
[J] Ce composant est un téléphone Android qui prend dans le composant [G] (getLocalisations) toutes ou une partie de la localisation et des images des objets suspects.
[K] Ce composant contient tous les éléments de cartographies des pistes du ou des terrains d'aviation qui sont gérés par l'IHM d'administration.

Question 2 :

Faire l'Architecture **TECHNIQUE** de ce Système d'Information en expliquant clairement comment vous pensez réaliser les composants identifiés lors de la question 1 (choix techniques, type d'architecture, …) et leurs décompositions éventuelles.

On utilise la technologie JEE pour réaliser le SI.

Le composant [C] est un JMS Provider contenant 2 canaux d'évènement en mode Topic. Un pour notifier les triplets d'image de [B] aux [D], et un autre pour notifier les triplets d'images verrouillées d'un [D] vers les [B].

Les IHM VISU et le téléphone Android sont des navigateurs et/ou interface Web qui communiquent avec :

- un Servlet pour aller chercher les images qui ont été notifiées dans un EJB Statefull via un DrivenMessage,
- et un autre Servlet qui est utilisé pour ajouter les localisations dans le composant [G].

L'IHM Admin est une IHM standalone qui communique directement avec les composants [E] et [F].

Les composants [A],[E] et[F] sont tous des EJB Statefull. Il y en a un de chaque pour chacun des robots.

Le composant [G] est une EJB Stateless qui :

- met en persistance les données via un Entity Localisation qui stocke les localisations dans une base de données
- lit avec la persistance les données via un Entity Localisation depuis la base de données.
-

Le composant [K] est une table dans la base de données. Le composant [H] réalise les requêtes dans la base de données.

Le robot est une JVM embarquée qui pilote le robot et communique avec une connexion URL avec un Servlet qui sert d'interface de communication avec les composants [B],[E] et [F].

Fin du sujet