

Module : Projet fédéré

TP n°2 : Mise en place de l'outil « Enterprise Architect (EA) »

I. Objectif du TP :

L'objectif de ce TP est de se familiariser avec l'interface utilisateur graphique de l'outil « **Enterprise Architect** » permettant d'éditer un ensemble de diagrammes du langage UML tel que : le *diagramme de cas d'utilisation*, *diagramme de classe*, *diagramme de séquence*, *diagramme d'états-transition* et *diagramme d'activités*.

II. Découvrir l'outil « Enterprise Architect (EA) » :

« Enterprise Architect » est un **outil graphique** conçu pour aider les équipes de développement de projets à construire des systèmes robustes et maintenables. À l'aide de rapports et de documents intégrés de haute qualité, des vues partagées sont facilement fournies, permettant la collaboration et renforçant la confiance dans les projets les plus complexes.

C'est un logiciel de modélisation et de conception UML publié par la société australienne « Sparx Systems ». C'est l'un des logiciels de conception et de modélisation les plus reconnus par ses capacités couvrant toutes les phases du cycle de conception des applications.

Etude de cas:

On désire réaliser une application pour la gestion des Rapports Quotidiens de Vol (RQV) de véhicules dans les départements de police, via le web.

On distingue initialement deux types d'utilisateurs pour ce système : les **victimes** et les **témoins**. Chacun de ces utilisateurs peut créer une déclaration de vol, en y indiquant son rôle (victime, témoin ou bien les deux), ses informations personnelles (son n°CIN, nom, prénom, adresse, tél), le type de la propriété volée (véhicule à moteur ou bien bicyclette) ainsi que les différentes informations disponibles qui l'identifient (couleur, marque, numéro de série pour les bicyclettes, matricule pour les véhicules à moteur, description générale), la date, l'heure et le lieu (avec tous les détails disponibles : n° de la rue, ville, code postal,...) du vol.

Le système attribue à chaque déclaration un identifiant, que l'utilisateur l'utilise pour pouvoir éditer la déclaration (ajouter des informations, supprimer la déclaration), avant de sauvegarder la déclaration. Le système doit enregistrer, pour chaque déclaration, la date de sa dernière modification.

On distingue également un autre type d'utilisateurs : **l'agent policier** qui se charge de la création des Rapports Quotidiens de Vol. Un RQV est relatif à une date particulière, il contient toutes les déclarations de vols effectuées ou bien modifiées dans ce jour. Lorsqu'un véhicule déclaré est retrouvé, l'agent policier modifie l'état de la déclaration concernée. Évidemment, l'agent policier doit s'authentifier pour pouvoir accéder à cette application.

On désire déterminer pour chaque RQV la liste des nouvelles déclarations, la liste des déclarations mises à jour, ainsi que les déclarations qui ont été résolues.

Travail à faire :

Activité 1. Établir le **diagramme de cas d'utilisation**.

Activité 2. Établir le **diagramme de classes** de ce système.

Activité 3. Établir le **diagramme de séquence** de la fonctionnalité suivante « Créer déclaration de vol ».

Activité 1: Création de diagramme de cas d'utilisation

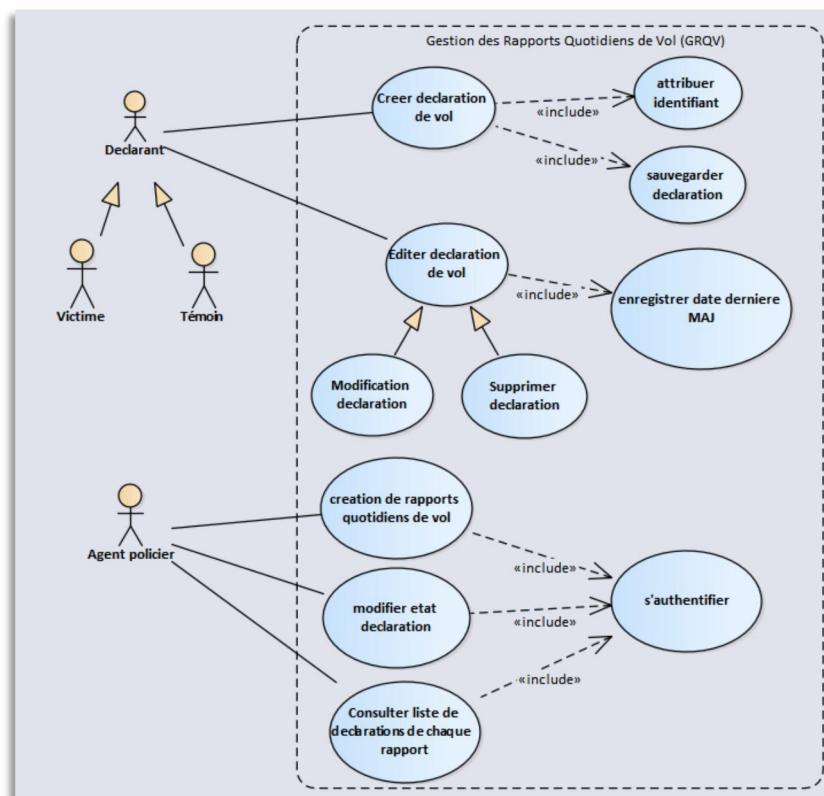
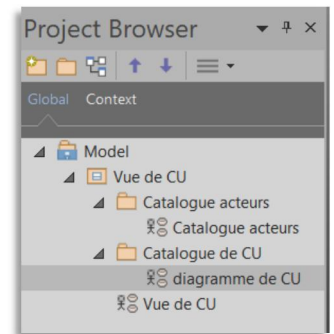


Figure 1 . Diagramme de cas d'utilisation de GRQV

1. Création d'une structure arborescente de vue de cas d'utilisation :

✧ Créer un **nouveau projet**.

- ✧ Créer une **vue de cas d'utilisation** nommée «Vue de CU».
- ✧ Ajouter un **sous-répertoire** (package) représentant le catalogue d'acteurs nommé « Catalogue acteurs » et sous ce répertoire ajouter un diagramme de CU portant le même nom que celui du sous-répertoire.
- ✧ Ajouter un **sous-répertoire** représentant le catalogue de cas d'utilisation nommé « Catalogue de CU » et sous ce répertoire un diagramme de CU portant le nom « Diagramme de CU ».

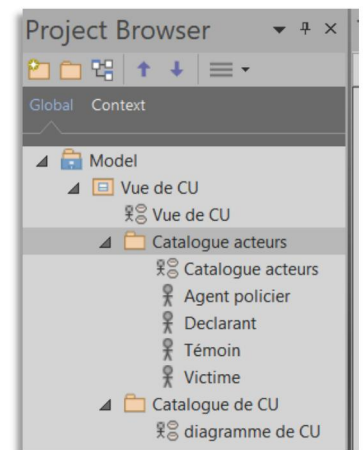


2. Ajouter un sujet dans le diagramme de cas d'utilisation :

- ✧ Dans le diagramme de CU ajouter le composant « **Boundary** » désignant les limites du système.
- ✧ Nommer le boundarub « Gestion des Rapports Quotidiens de Vol (GRQV) ».

3. Ajouter les acteurs dans le diagramme de cas d'utilisation.

- ✧ Glisser les acteurs (*Declarant*, *Victime*, *Temoin* et *Agent policier*) de votre diagramme de CU.
- ✧ Dans le menu «Project Browser», glisser tous les acteurs sous le répertoire « Catalogue acteurs ».

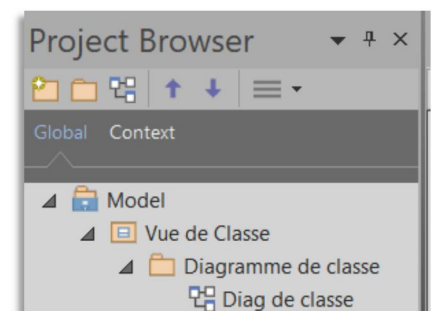


4. Création d'associations entre acteurs et diagrammes.

- ✧ Glisser les cas d'utilisations et créer les relations nécessaires en se référant au diagramme de cas d'utilisation de la Figure 1.

Activité 2: Création de diagramme de classe du système

- ✧ Créer une **vue de diagramme de classe** nommée «Vue de Classe».
- ✧ Ajouter un **sous-répertoire** (package) nommé « Diagramme de classe » et sous ce répertoire ajouter un diagramme de classe nommé « Diag de classe ».
- ✧ Ajouter dans « Diag de classe » les classes décrites dans la Figure 2 ainsi que les différentes relations.



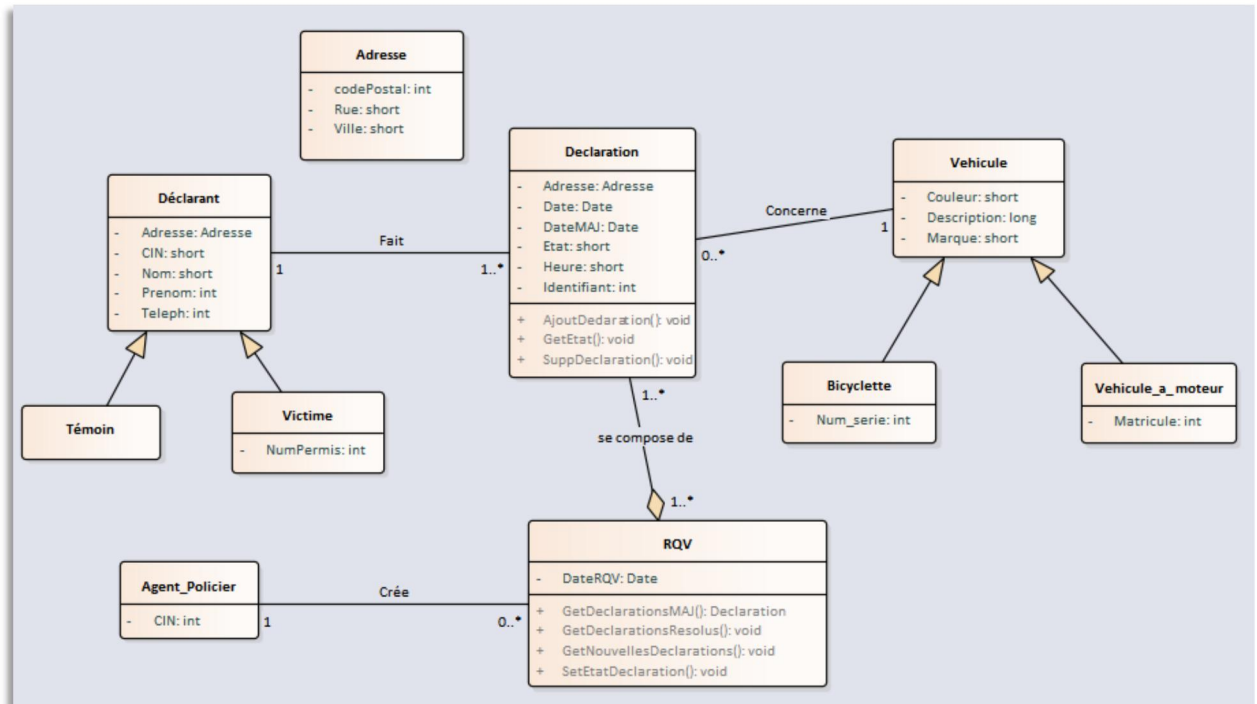
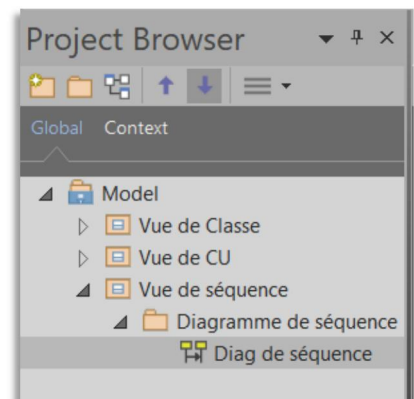


Figure 2. Diagramme de classe

Activité 3: Création de diagramme de séquence de la fonctionnalité suivante « Créer déclaration de vol ».

- ✧ Créer une **vue de diagramme de séquence** nommée «Vue de séquence».
- ✧ Ajouter un **sous-répertoire** nommé « Diagramme de séquence » et sous ce répertoire ajouter un diagramme de séquence et nommer le « Diag de séquence ».



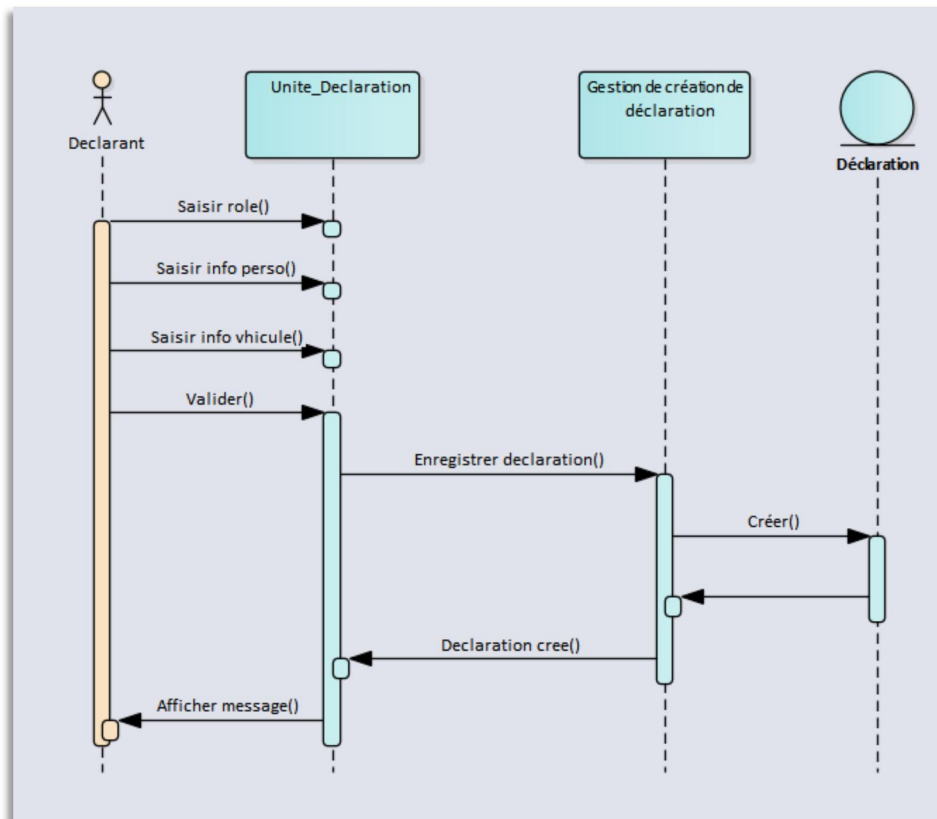


Figure 3. Diagramme séquence du CU « Créer déclaration de vol »