

# Projet V2.9.0

---



Cette œuvre est mise à disposition selon les termes de la [licence Creative Commons Attribution – Pas d'Utilisation Commerciale – Partage à l'Identique 3.0 non transposé](#).

Document en ligne : [www.mickael-martin-nevot.com](http://www.mickael-martin-nevot.com)

---

Date de rendu : **04 novembre 2018 à minuit**  
Date de validation du sujet : **12 octobre 2018 à 17h30**  
Travail : **en binôme**

## 1 Assistance

Vous pouvez contacter l'enseignant en cas de besoin en formalisant et en ciblant précisément votre demande. Pour ce faire, vous devez respecter les règles de communication et d'envoi (ci-dessous).

## 2 Communication et envoi

### 2.1 Généralités

En joignant vos coordonnées (*e-mail* et téléphone portable notamment) à un message ou à votre livraison, vous pourrez être joint en cas de problème.

### 2.2 Communication

Chaque communication devra être faite :

- à l'adresse électronique : [mmartin.nevot@gmail.com](mailto:mmartin.nevot@gmail.com) ;
- en faisant figurer [AMU] [LP] en début de sujet.

### 2.3 AMeTICE

#### 2.3.1 Présentation

Pour votre projet, vous devez utiliser la plateforme pédagogique de l'Université d'Aix-Marseille : **AMeTICE** (<http://ametice.univ-amu.fr>), à l'adresse suivante : Accueil / Cours / IUT / LP – Sciences et Technologies / Web, e-commerce et big data (WEB)" – Aix-en-Provence / [18]-S1 – ECUE14 Algorithmique et UML (M. Martin-Nevot).

#### 2.3.2 Sélection du groupe

Sélectionnez, sur AMeTICE, un **groupe** d'étudiant disponible avec votre binôme.

#### 2.3.3 Livraison

Votre livrable devra être :

- nommé de la manière suivante (**Nom1**, **Nom2** étant vos noms et **Prénom1**, **Prénom2** vos prénoms) : **Nom1 Prénom1 – Nom2 Prénom2** ;
- compressé dans une seule archive au format ZIP n'excédant pas **20 Mo** ;
- remis, avant la date de rendu, sur AMeTICE.

### 3 Sujet

Vous devez écrire la **spécification fonctionnelle technique** et coder l'**implémentation** d'une application en **Java** qui doit utiliser les concepts vus lors de cet enseignement.

Le sujet est libre mais devra être basé sur les réalisations des TD TD4–2 : Java, cas pratique, TD5 : Algorithmique « avancée » et TD6 : UML. Pensez à une réalisation qui vous tienne à cœur : simulation d'un jeu vidéo connu, gestion d'un parc d'attraction, d'un zoo, d'une entreprise de restauration, élevage (animal réel ou virtuel), etc. mais veillez tout de même à respecter scrupuleusement les consignes de ce document. **La description du projet devra être approuvée par l'enseignant, durant les séances de TD, avant la date de validation du sujet** (prévoyez que votre description puisse être acceptée après plusieurs propositions).

La partie « graphique » de votre application peut être minimaliste (affichage en ligne de commande par exemple).

Vous devez utiliser le plus possible d'**algorithmique**, de **structures de données**, de particularité du langage **Java**, de **modèles de conception** et de **diagrammes UML** vus dans cet enseignement. En particulier, votre application doit comporter des **classes abstraites**, des **interfaces**, des **collections**, au moins un algorithme de **tri** et utiliser le modèle **MVC**. De plus, vous devez réaliser l'ensemble des diagrammes de **cas d'utilisation**, de **séquences** et de **classes** de l'application.

Vous devez apporter un soin tout particulier à la présentation du **code source** (**indentation**, respect d'une **convention de nommage**, **commentaires**, etc.) et à l'**architecture** des répertoires.

Votre application doit être **évolutive**, **modulaire** et professionnelle (**robuste**, **fiable** et intégralement **fonctionnelle**).

### 4 Livrables

Vous devez envoyer les éléments suivants avant la date de rendu :

- la **spécification** minimale (sous forme d'un ou plusieurs documents) :
  - la **description** (validée) de l'application ;
  - l'**étude de conception** ;
  - les **diagrammes UML** ;
  - la clôture du travail (**synthèse en anglais**, **bilan technique**, **problèmes rencontrés**, **écarts avec les prévisions**, **mesures d'amélioration**) ;
- l'**application** :
  - le **code source** ;
  - l'application au format **JAR** ;
  - le **projet Eclipse** (si vous utilisez Eclipse dans le cadre du projet) ;
  - le fichier « **lisez-moi** » ;

- le **manuel d'utilisation** (présentant plusieurs scénarios d'utilisation pas à pas) ;
- la documentation complète générée par la **Javadoc** de l'application.

## 5 Conseils

Voici quelques conseils :

- pensez à utiliser des algorithmes en **pseudo-code** pour formaliser vos idées avant de coder des algorithmes complexes ;
- gérez la répartition de réalisation des fonctionnalités entre les membres du groupe (en se répartissant les différents modules par exemple) ;
- optimisez votre code source en tentant de diminuer la complexité de vos algorithmes ;
- testez rigoureusement votre application.