

**PlanIFTicateur**

**Travail pratique 2**

GLO-2004 – Génie Logiciel Orienté Objet – Hiver 2015

Travail présenté à

M. Jonathan Gaudreault

Chayer, Philippe [Philippe.chayer.1@ulaval.ca](mailto:Philippe.chayer.1@ulaval.ca) IFT PHCHA47

Khouma, Abdou [abdou.khouma.1@ulaval.ca](mailto:abdou.khouma.1@ulaval.ca) GIF ABKHO9

Gadoury, Gabriel [Gabriel.gadoury.1@ulaval.ca](mailto:Gabriel.gadoury.1@ulaval.ca) IFT GAGAD1

Yeo, Clotioloman [Clotioloman.yeo.1@ulaval.ca](mailto:Clotioloman.yeo.1@ulaval.ca) GLO CLYEO1

Table des matières

[Introduction 3](#_Toc416100361)

[Diagramme des classes de conception 4](#_Toc416100362)

[Diagramme de package 6](#_Toc416100363)

[Diagrammes de séquence 7](#_Toc416100364)

[Indicateur du nombre de cours par jour 7](#_Toc416100365)

[Affichage de la grille horaire 9](#_Toc416100366)

[Sélection d’un cours dans la grille horaire avec l’aide de la souris / déplacement cours dans la grille horaire 10](#_Toc416100367)

[Validation qu’un jour / heure de début est valide pour un cours donné 12](#_Toc416100368)

[Annexe 14](#_Toc416100369)

[Modèle du domaine 14](#_Toc416100370)

[Modèle des cas d’utilisation 15](#_Toc416100371)

[Glossaire 16](#_Toc416100372)

[Gestion de projet 20](#_Toc416100373)

# Introduction

La construction des horaires de session est une tâche difficile pour les directions de programme. À l’aide du logiciel dont nous avons le mandat, cette tâche deviendra visuelle et interactive, donc plus facile à réaliser. Ainsi, la direction économisera beaucoup de temps et, par le fait même, diminuera significativement le coût associé à la production d’un horaire optimal.

Ce rapport consiste à présenter le modèle de conception mis à jour et l’architecture logicielle de planIFTicateur.

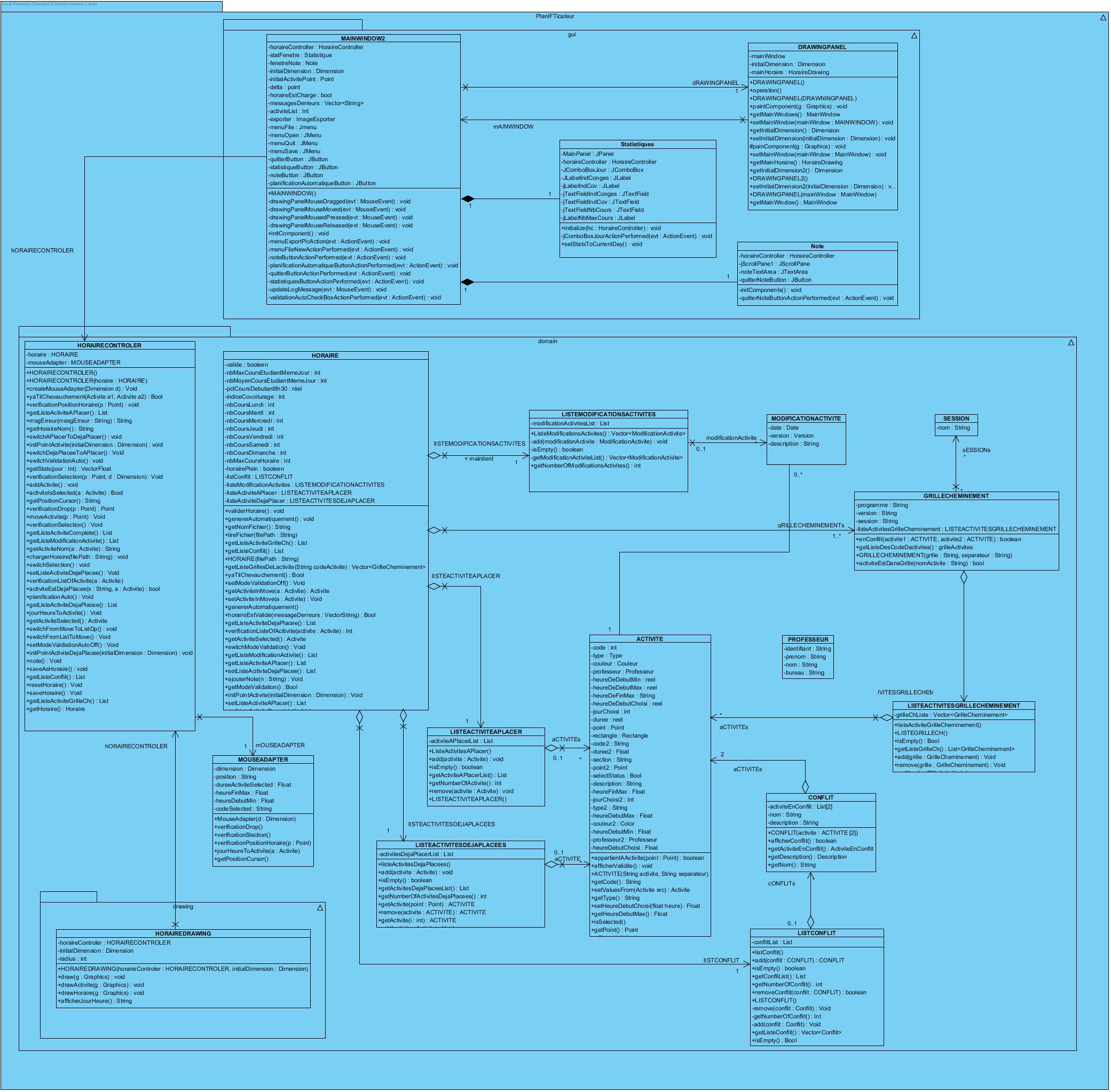
La première section montre un diagramme de classe. Il s’agit de l’architecture qui est utilisée afin d’implanter notre solution sous le langage Java. Ensuite, une section est consacrée à illustrer les différents diagrammes d’états de certains éléments de notre programme. Nous poursuivrons en détaillant les diagrammes de séquence qui permettront de mieux visualiser les fonctionnements particuliers de PlanIFTicateur et une dernière section mettra à jour la gestion du projet ainsi que les éléments du rapport précédent.

# Diagramme des classes de conception

Le diagramme qui suit représente l’état actuel des classes utilisées par notre application. Il y a beaucoup de changements qui ont été apportés. Tout d’abord, nous avons ajouté plusieurs méthodes et attributs dans les différentes classes. Certains attributs avaient été omis et nous les avons ajoutés. Nous avons aussi ajouté une classe **MouseAdapter** qui permet de faire certaines vérifications au niveau de la position de la souris pour le mouvement **Drag-and-Drop** au niveau de l’horaire. Notre contrôleur a aussi subit beaucoup de modification. Le contrôleur (ici **HoraireController**) est la seule classe qui échange avec l’interface. Tout élément de l’interface qui doit envoyer des informations passe par celui-ci. La même chose se produit lorsque les éléments du domaine désirent afficher sur l’interface utilisateur.

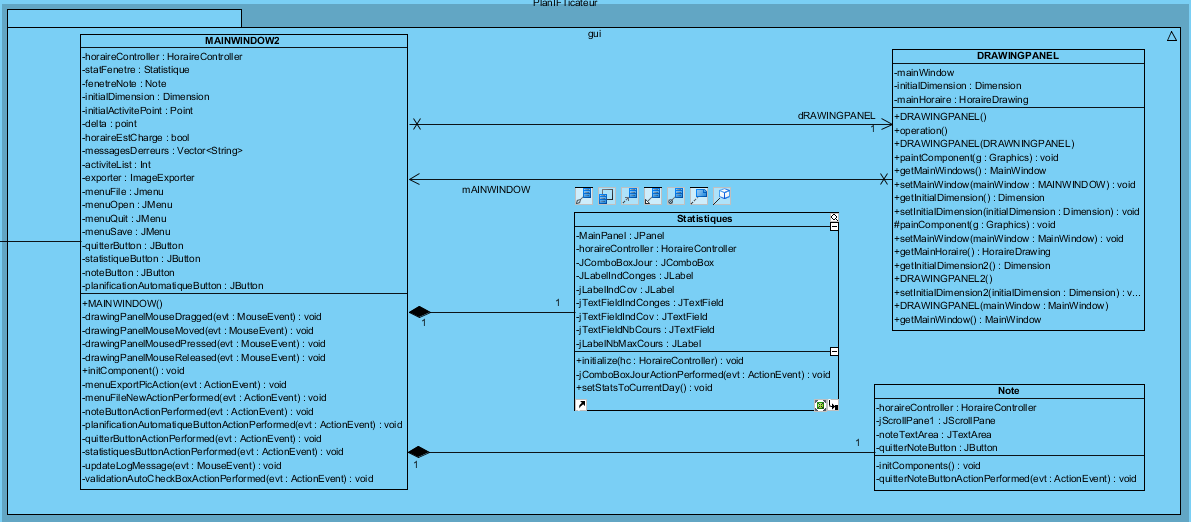
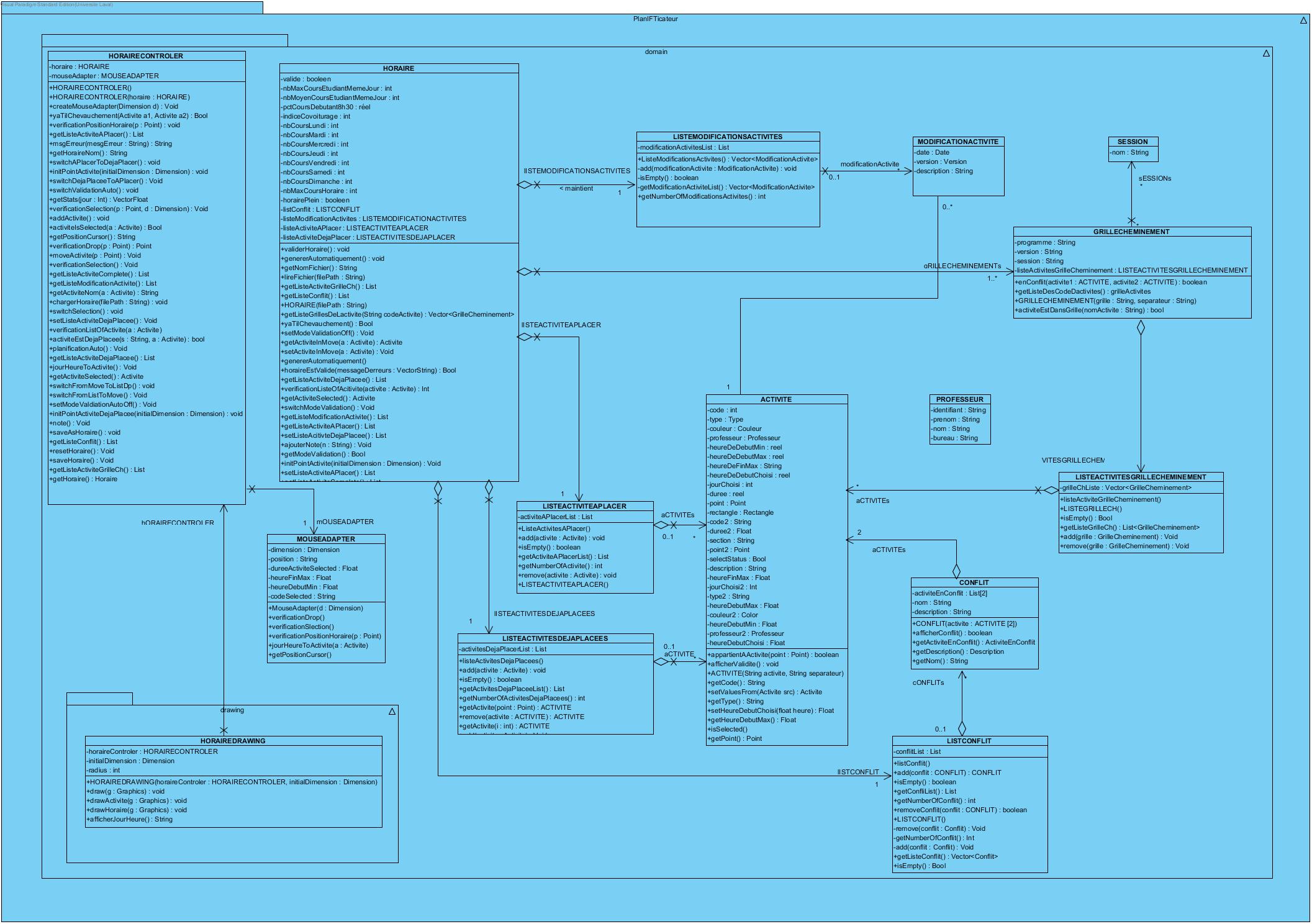
Au niveau de l’interface (**GUI**), nous avons ajouté une fenêtre de note et une fenêtre statistique. Les deux fenêtres additionnelles sont liées à la fenêtre principale (**MainWindow**) et sont automatiquement détruites lorsque la fenêtre principale est détruite.

**(Schéma sur page suivante.)**



Étant donné la grande taille du schéma, nous avons séparé le schéma du domaine et de l’interface pour permettre une lecture plus facile des différentes parties.

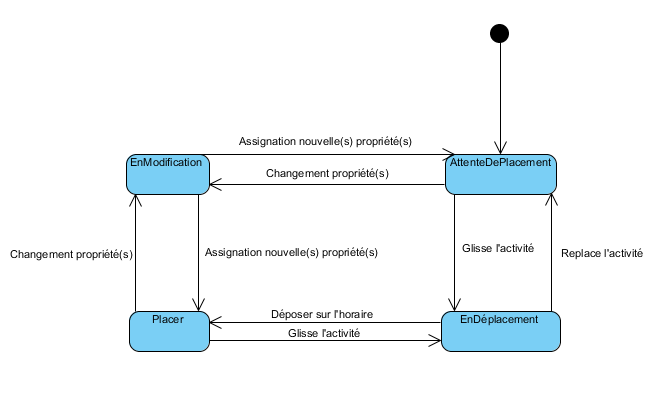
**(Il est aussi possible de le voir plus facilement à l’aide de visual paradigm. Voir fichier joint.)**



# 

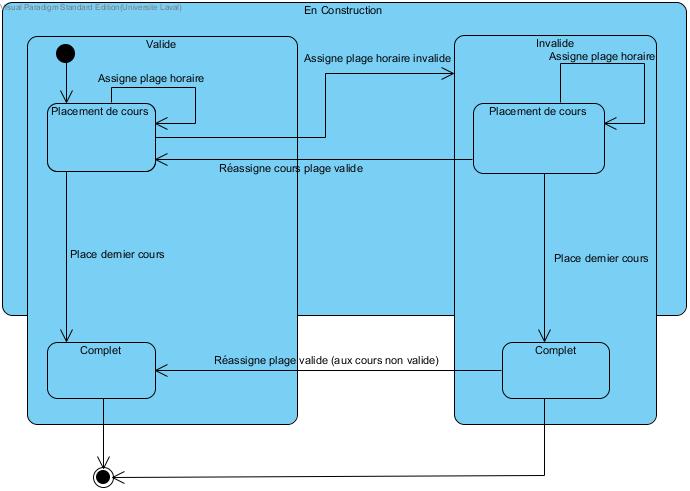
# Diagramme d’états

## Diagramme d’états d’une activité



Dans ce diagramme, nous pouvons voir les différents états d’une activité lorsque le programme est en fonction et que l’activité a été créée. L’état de départ est que l’activité est en attente d’être placée. Par la suite, il est possible de modifier certains champs d’une activité et de lui assigner de nouvelles propriétés. Il est possible de faire cela lorsque l’activité est placée sur l’horaire ou bien lorsqu’elle n’est pas encore placée (en déplacement). Lorsque l’activité est sélectionnée, elle est en déplacement (cet état est montré par le changement de couleur de l’activité). L’activité peut aussi avoir été placée sur l’horaire et remise dans la liste d’activités en attente d’être assignée à une plage horaire, encore une fois, en passant par l’état en cours d’être déposé.

## Diagramme d’états d’un horaire



Dans ce diagramme d’état, on peut constater que l’horaire peut être dans plusieurs états simultanément. Tout d’abord, lors que l’on commence l’horaire, il est en construction. Un horaire est soit valide, ou il ne l’est pas. Dans notre cas, un horaire est valide lorsqu’il n’y a pas de conflit d’horaire pour les cours qui appartiennent à la même grille de cheminement. Nous avons fait ce choix, puisque l’indicateur de validité prend en considération seulement les conflits pour afficher un horaire qui est valide. Cependant, un horaire que l’on doit considérer valide est un horaire qui est complet, sans conflit d’horaire et dont tous les cours sont placés. Cependant, l’indicateur ne représente pas cela, puisqu’il est plus utile de l’utiliser pour éviter les conflits d’horaire. En effet, si on le prenait pour afficher un horaire complètement valide, l’indicateur serait rouge seulement lorsqu’on placerait le dernier cours et qu’il n’y avait pas de conflit, ce qui dans notre cas, ne vaut pas la peine. L’aide apportée par l’indicateur de conflit apporte beaucoup plus de valeur à notre application.

# Diagrammes de séquence

## 

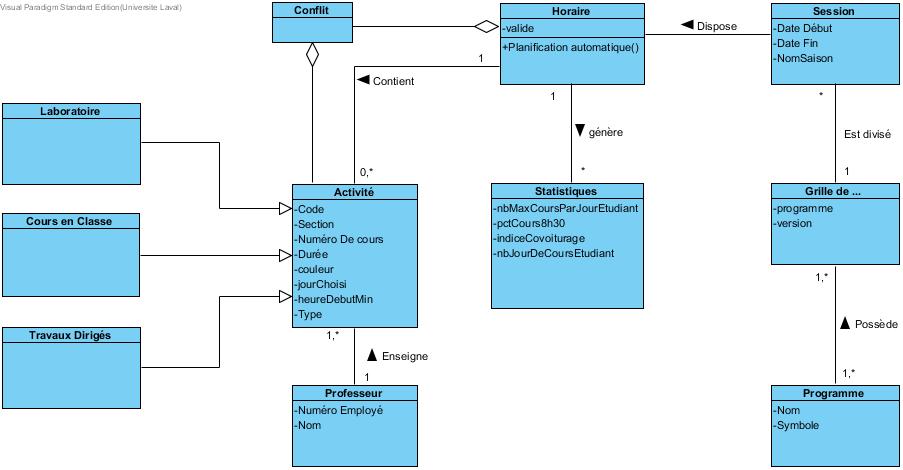
## Indicateur du nombre de cours par jour

## Affichage de la grille horaire

## Sélection d’un cours dans la grille horaire avec l’aide de la souris / déplacement cours dans la grille horaire

# Annexe

## Modèle du domaine



## 

## diagramme usecaseModèle des cas d’utilisation

## Glossaire

**Algorithme**

Suite de règles permettant de résoudre un problème.

**Application**

Programme ou ensemble de programmes visant à aider un utilisateur d’un ordinateur dans le traitement d’une tâche précise. *(Réf. : Larousse.fr)*

**Cas d’utilisation**

Relation établie entre une fonctionnalité et, soit un acteur ou une autre fonctionnalité.

**Chemin**

Emplacement précis sur un disque dur définit par la suite ordonnée des dossiers pour avoir accès au fichier recherché.

**Diagramme de séquence système (DSS)**

Représentation des interactions chronologiques entre un système et ses acteurs. *(réf. : Wikipédia)*

**Domaine d’affaires**

Réfère pour l’ensemble des processus d’affaires d’un projet ou d’une organisation. Englobe les entités, les acteurs et autres participants de ces processus.

**« Drag and drop »**

***(Terme anglais)*** En informatique, processus durant lequel un usager d’un ordinateur sélectionne un objet, au moyen d’une souris, en maintenant enfoncé le bouton gauche de cette dernière afin de déplacer le dît objet et de le relâcher à l’endroit voulu en relâchant également le bouton de sa souris.

**Fenêtre**

Zone d’affichage d’information d’un programme.

**Fonction**

Bloc d’une séquence d’instructions visant un but, une fonctionnalité précise.

**Logiciel**

Ensemble d’instructions et de règles interprétables par un ordinateur.

**Multiplateforme**

Fonctionnant sur plusieurs plateformes, soit plusieurs ordinateurs/systèmes d’exploitation différents.

**Multiutilisateur**

Offrant la possibilité à plusieurs usagers d’interagir simultanément sur le même logiciel.

**PlanIFTicateur**

Nom du logiciel en développement pour le projet contenu dans ce document.

**Plateforme PC**

Liaison entre un ordinateur personnel munit d’un processeur spécifique et du système d’exploitation *Windows*.

**Presse-papier**

Fonction intégrée dans tous les systèmes d’exploitation stockant des données que l’on souhaite déplacer ou copier.

**Programme**

Succession d’instructions qu’un ordinateur peut exécuter afin d’accomplir des opérations.

**Serveur dédié**

Système informatique dont l’ensemble des ressources est dédié à un seul utilisateur.

**Serveur web**

Système informatique qui a pour fonction la publication de sites web à la demande d’un autre système.

**Session**

Période de 15 semaines durant laquelle un étudiant de l’université doit accomplir les objectifs de ses cours.

**GUI**

Signifie : **G**raphical **U**ser **I**nterface, ce qui représente l’interface graphique utilisateur. C’est ce qui est affiché par le programme.

**Diagramme de package**

Permet de représenter l’architecture des différents groupements de classe en différentes couches logiques.

**Diagramme de séquence**

Diagramme illustrant ce qui se produit lors de certaines actions. Permet de comprendre la communication entre les classes.

## Gestion de projet

***Voir diagrammeDeGant.gan***