

Rapport Technique

Florian Barrois Nicolas Devillers Valentin Jeanroy Mehdi Loisel
Jean Mercadier Ismail Taleb Willeme Verdeaux

21 mars 2014

Table des matières

1	Installation	3
1.1	Installation	3
1.1.1	Prérequis	3
1.1.2	Obtenir l'application	3
2	Structure	4
2.1	MVC	4
2.1.1	Modele	4
2.1.2	Vue	4
2.1.3	Controleur	4
3	Modelisation	5
3.1	UML	5
3.2	Base de donnée	5

Introduction

Chapitre 1

Installation

1.1 Installation

1.1.1 Prérequis

Pour lancer l'application, il est obligatoire d'avoir à disposition un server MySQL. Ainsi au lancement le programme chargera ses information dans la base de donné. Mais si la base de donné n'existe pas celui sera crée avec le nom de stcal.

Cette base de donnée permet ainsi à l'application de sauvegarder et recharger ses données au lancement et à la fin de l'exécution du programme.

note : Il est conseillé de créer un utilisateur spécifique au programme qui aura tous les privilèges sur la base de données stcal.

1.1.2 Obtenir l'application

L'intégralité de l'application se trouve sur GitHub sur le repo stcal : <https://github.com/Ricain/stcal>

Executable Pour obtenir un exécutable de l'application, soit un fichier jar il suffit de le télécharger sur GitHub à l'URL suivant : <https://raw.githubusercontent.com/Ricain/stcal/Main/stcal.jar>

Code source Pour obtenir le code source d'une manière "propre" (sans le télécharger directement dans un fichier zip sur GitHub), il suffit de cloner le projet :

```
$ git clone https://github.com/Ricain/stcal
```

L'intégralité du projet sera cloné dans un répertoire stcal sous le répertoire courant. Cela nécessite d'avoir git installé. Dans un environnement linux, le gestionnaire de paquet permet de l'installer. Sur un mac, il faut installer et exécuter la commande suivante après avoir installé Xcode :

```
$ xcode-select -install
```

Une fois git installé il est possible de faire des "commit" et des "pull request".

IDEA Il est cependant possible de cloner directement le projet à partir d'IntelliJ IDEA. Pour cela il suffit d'importer un projet à partir d'un VCS (Version Control System). L'IDE vous demandera le lien GitHub de l'application énoncé plus haut.

Chapitre 2

Structure

2.1 MVC

L'arborescence du projet a été pensée pour respecter au mieux le modèle vue contrôleur. Sous le répertoire *src* on trouve trois autres répertoires :

control contient l'ensemble des classes moteur à l'application (contrôleur).

fen abréviation de fenêtre, contient l'ensemble des classes permettant de dessiner l'application (vue).

don abréviation de données, contient l'ensemble des classes concernant les données de l'application (modèle).

2.1.1 Modèle

Le modèle constitue les données de l'application. L'ensemble des classes prévues à cet effet sont stockées sous le répertoire *don*. On y trouve des classes comme :

- Etudiant
- Soutenance
- Prof
- Agenda
- etc

Sauvegarder les données de l'application permet non seulement de garder un historique des stages mais également de réouvrir l'application et de la retrouver tel qu'on l'a fermée. À cet effet on trouve un sous-répertoire *manager* qui contient un ensemble de classes permettant de faire le lien entre la base de données et le modèle de données de l'application. Chaque manager **doit** implémenter l'interface *manager* afin de respecter le concept de JBDD (enseigné en S4).

Le script qui permet d'installer la base de données se trouve dans le répertoire *res* (ressource).

Les classes constituant le répertoire *don* ont été pensées sur un modèle bien précis. On verra le MCD et l'UML dans la partie modélisation de ce rapport.

2.1.2 Vue

La vue est constituée des classes sous le répertoire *fen* elles permettent de dessiner les fenêtres. La bibliothèque graphique utilisée ici est *Swing*.

La fenêtre principale est dessinée par la classe *FInterface*. On lui ajoute des objets de type *FTab* afin de créer des onglets. Il est important de noter que aucune des classes fenêtres (classes dans le répertoire *fen*) n'hérite d'un quelconque objet de la bibliothèque graphique, elles sont plutôt composées de ces objets.

La classe *Main* a une méthode *mac*. Cette méthode est importante pour adapter la partie graphique au système OS X.

2.1.3 Contrôleur

Chapitre 3

Modelisation

3.1 UML

3.2 Base de donnée

Conclusion