

# Contents

0.1	Server . . . . .	1
0.2	Scheduler . . . . .	1
0.3	HeaterManager . . . . .	1
0.4	DatabaseAccess . . . . .	2
0.5	Listen_thread . . . . .	2
0.6	Update_daemon . . . . .	2
0.7	DayPlanner_daemon . . . . .	2
0.8	Day . . . . .	2

## 0.1 Server

Le serveur instancie un objet *Scheduler* et gère les communications client-serveur. Pour chaque communication, un thread de la classe *Listen\_thread* est instancié et lancé. Un thread *DayPlanner\_daemon* est lancé pour déterminer l'horaire de la journée.

## 0.2 Scheduler

La classe se chargeant d'effectuer les changements de température et d'écrire dans la base de données.

Chaque jour, le scheduler reçoit de la part du thread incluant le Machine Learning un objet de la classe *Day* correspondant à l'horaire de la journée. Le scheduler contient un objet *HeaterManager* pour pouvoir modifier la température des pièces de la maison et un objet *DatabaseAccess* pour écrire dans la base de données les changements de température au fur et à mesure de la journée.

Remarque : c'est dans cette classe que doit être géré le temps nécessaire au chauffage d'une pièce.

## 0.3 HeaterManager

Cette classe sert à faire le lien entre le hardware de la maison et le programme, elle reçoit des ordres concernant la température des pièces, elle ne

gère ni l'heure ni le temps nécessaire à chauffer la pièce.

## **0.4 DatabaseAccess**

Sert d'interface entre la database et le programme.

## **0.5 Listen\_\_thread**

Instancié à chaque connection entrante par le serveur, cette classe sert à gérer les requêtes des clients.

## **0.6 Update\_\_daemon**

Daemon se chargeant de changer les températures via le scheduler au bon moment.

## **0.7 DayPlanner\_\_daemon**

Ce daemon contiendra toute la partie "Machine Learning" du programme. Il se réveillera une fois par jour pour déterminer l'horaire de la journée et le fournir au scheduler.

## **0.8 Day**

Une ADT servant à contenir l'horaire d'une journée.