FAS Class Scheduler Documentation

Table of Contents

[2 Introduction 1](#_Toc436734171)

[3 Features 1](#_Toc436734172)

[4 contribute 1](#_Toc436734173)

[5 Detailed Description 1](#_Toc436734174)

[6 developers and Support 5](#_Toc436734175)

# Introduction

The Class Scheduler is a well needed software at the faculty of arts and sciences at the American University of Beirut in particular, and in any academic department in general. It finds the best fit solution to schedule classes in rooms based on variables, including room availability, capacity, accessories, professor’s availability and more.

This program has two parts: front end, takes input from departments at the university. Back end, finds the best combination possible among the classes requested by the departments.

# Features

* Processing data from the database happens with the help of the open source code “MyBatis”, a software that facilitates the creation of java objects based on the database fields in a neat and efficient way.
* The objects created are then stored in the necessary datatypes (arrays, maps, sets, vectors etc...) to be accessed easily afterwards.
* When all the necessary data is set, it gets passed to one of two solving algorithms: solve by room or solve by time.
* Using the first algorithm, the program starts by allocating a room as a first step, and then places a class in it, based on the priority of the department requesting the room and the fitness of the room in general. If a certain request could not be granted, the room stays the same and the timing changes.
* Using the second algorithm, the program keeps the same time requested fixed and picks a different room having a location close to the requested one.
* After finding the perfect fit, the solution gets outputted in any format requested by the user.

# contribute

MyBatis-3: https://github.com/mybatis/mybatis-3

# Detailed Description

Main package name: src/main/java/lb/edu/aub/cmps/

* Sub-package name: algorithm
  + Class: BasicScheduler.java
  + Description: Extends Scheduler Implements IScheduler. Classes that could not be scheduled perfectly (as requested by department), find their second best choice in the method “schedule ( )”.
  + Class: ByTimeScheduler.java
  + Description: Extends Scheduler Implements IScheduler. Starts with a specific time and schedules all the classes in their best or second best rooms.
  + Class: IScheduler.java
  + Description: An interface with a public method schedule ( ).
  + Class: Runner.java
  + Description: The program’s entry point. Main method exists here.
  + Class: Scheduler.java
  + Description: Abstract class implementing IScheduler. Defines the object Scheduler.
  + Class: SetUp.java
  + Description: A class that retrieves all the info from the database and stores them in the necessary datatypes. It calls on object services from package “services”, so that they can be used in finding the solution later in the Schedulers.
* Sub-package name: classes
  + Class: Building.java
  + Description: Defines the object Building.
  + Class: Class.java
  + Description: Defines the object Class.
  + Class: ClassTimeComparator.java
  + Description: Implements Comparator. Compares the times of 2 given Classes.
  + Class: Course.java
  + Description: Defines the object Course.
  + Class: Day.java
  + Description: Defines the enumerator Day.
  + Class: Department.java
  + Description: Defines the object Department.
  + Class: DepartmentWeightComparator.java
  + Description: Implements Comparator. Compares the weights of 2 given Departments.
  + Class: MyLogger.java
  + Description: Defines the logging system, logging file and the necessary methods.
  + Class: PrintTimeClassInRoomVisitor.java
  + Description: Prints the resulting output of the scheduling algorithm to an output file.
  + Class: Professor.java
  + Description: Defines the object Professor.
  + Class: Room.java
  + Description: Defines the object Room.
  + Class: RoomByTimeSlotComparator.java
  + Description: Implements Comparator. Compares the times of 2 given Rooms.
  + Class: Roomvisitable.java
  + Description: Interface with public method accept (RoomVisitor room).
  + Class: RoomVisitor.java
  + Description: Interface with public method accept (Room room).
  + Class: Time.java
  + Description: Defines the object Time.
  + Class: TimeSlot.java
  + Description: Defines the object TimeSlot.
* Sub-package name: grad
  + Class: App.java
  + Description:
* Sub-package name: mappers
  + Class: AccessoryMapper.java/xml
  + Description: Creates a mapper between the Accessory and the database.
  + Class: BuildingMapper.java/xml
  + Description: Defines the necessary mappers between the class Building and the database, in order to retrieve all the necessary information.
  + Class: ClassMapper.java/xml
  + Description: Defines the necessary mappers between the class Class and the database, in order to retrieve all the necessary information.
  + Class: CourseMapper.java/xml
  + Description: Defines the necessary mappers between the class Course and the database, in order to retrieve all the necessary information.
  + Class: DepartmentMapper.java/xml
  + Description: Defines the necessary mappers between the class Department and the database, in order to retrieve all the necessary information.
  + Class: ProfessorMapper.java/xml
  + Description: Defines the necessary mappers between the class Professor and the database, in order to retrieve all the necessary information.
  + Class: RoomMapper.java/xml
  + Description: Defines the necessary mappers between the class Room and the database, in order to retrieve all the necessary information.
* Sub-package name: output
  + Class: GenerateNotScheduledInfo.java
  + Description: Loops through all the not scheduled classes and prints them and the cause they couldn't be scheduled.
* Sub-package name: services
  + Class: AccessoryService.java
  + Description: Uses the mappers to build a set of the accessories used.
  + Class: BuildingService.java
  + Description: Uses the mappers to build a set of the buildings saved in the database.
  + Class: ClassService.java
  + Description: Uses the mappers to build a set of the classes saved in the database, with all its details: Professor, Classroom, Time etc…
  + Class: CourseService.java
  + Description: Uses the mappers to build a set of the courses found in the database, with their corresponding classes.
  + Class: DepartmentService.java
  + Description: Uses the mappers to build a set of the departments saved in the database, with their corresponding offered courses.
  + Class: MyBatisUtil.java
  + Description: Defines MyBatis configuration file.
  + Class: ProfessorService.java
  + Description: Uses the mappers to build a set of the professors saved in the database, with all the classes each one is giving.
  + Class: RoomService.java
  + Description: Uses the mappers to build a set of the rooms saved in the database, with their corresponding accessories.

Testing package name: src/test/java/lb/edu/aub/cmps/grad

* + Class: AppTest.java
  + Description: A class with test cases to insure code validity and cleanliness.

# developers and Support

Julia El Zini: [juliaelzini@gmail.com](mailto:juliaelzini@gmail.com)

Yasmin Kadah: [kadahyasmin@gmail.com](mailto:kadahyasmin@gmail.com)

Bilal Abi Farraj: [bilalabifarraj@gmail.com](mailto:bilalabifarraj@gmail.com)