# Classroom Scheduling

There are 2 main sections to this application; administration and generation. Administration would set the rules, classes, professors, rooms and other settings and configurations for the application. Generation would generate a schedule of professors teaching classes in given rooms given the rules and constraints.

## Administration

Some items would need to evolve from semester to semester, while others would mostly remain unchanged for longer periods of time.

Each semester, rules, classes, professors and other items may need to be copied from a previous semester.

Classes, professors, and rooms are fairly straight forward. They just need creat/edit/update/delete functionality. Classes may include the section for each class as an atomic item, or the class may have sub items to represent the sections.

The administrator will assign professors to teach certain classes, so the classroom scheduling will be combinatorial of Class/Professors X room X time.

### Rules

The rules will need to be much more dynamic. Since rules have to be very flexible, it may need its own rule language to be defined, or the design should include plug-ins to allow quick and easy addition of new situations. Either way proper design patterns should be employed.

Rules have varying degrees of granularity to be applied. A rule may be that Teacher X does not want to teach Class Z after 7pm on M/W/ or Fridays. The rule could also be relaxed to simply say that no teacher should teach any class after 7pm on M/W/F.

#### Types of rules

##### Intra Class/Teacher/Room Rules

Intra Class/Teacher/Room Rules are rules that do not involve other combinations of classes, teachers and rooms.

Examples of rules

* Teacher X does not want to teach on M/W/F before 4, or on T/TH after 5

##### Inter Class/Teacher/Room Rules

Inter Class/Teacher/Room Rules are rules that do not involve other combinations of classes, teachers and rooms.

Examples of rules

* Teacher Y cannot teach 3 classes back to back
* Class Z should not be at the same time as Class Y ( this may be a specific section or all sections )
* Class Z should follow Class Y (this may be a specific section or all sections)
* Teacher X and Teacher Y should teach on the same days or times preferable.

##### Rule weighting

Some rules can be relaxed if necessary, so a weighting system may need to be incorporated. An infinite weight might indicate the rule has to be enforced and no schedule can ever violate this rule. A low number would indicate that it’s not desirable, but we could allow this under some circumstances. When comparing different schedules, a lower total weighted value would therefore be the more appropriate schedule.

## Generate Schedule

When all the rules a given, the user can begin generating schedules. Since this will likely be a time-consuming process the scheduler should likely save the state of a given job. That way if the process needs to stop then it could restart where it left off. The administrator may want to be able to view schedules that are created as the process begins running.

Stretch Goals

* Distributed processing to speed the lookup through the search space
* Containerization. Being able to spin up 50 servers to work in a distributed manner could help solve the problem quickly.
* Multiple users with different views and responsibilities. Some users may only be able to view the completed schedule. Others may only be able to edit classes etc.