P1 : Timetable Management System for an Academic Institution

Review 4 Report

Team 7 - DB4C

Aathish Sivasubrahmanian [3122 21 5001 001]

Ayshwarya B [3122 21 5001 017]

Harshida S P [3122 21 5001 031]

26.08.2022

FUNDAMENTALS OF PROGRAMMING AND SOFTWARE DEVELOPMENT

ABSTRACT

A timetable management system is a project that is intended to provide greater assistance for an institution's workers. The timetable management system allows you to create and see particular staff and student timetables, as well as the lab timetables. This system also includes files that hold subject information for both staff and students. A timetable is a temporal organization of a group of personnel and students that ensures that all stated constraints are satisfied. As a result, we devised a realistic strategy for developing staff, student, and lab timetable systems that can be tailored to any institution's timetable. Once a semester's timetable is completed, any member of staff and student can access it. The schedule module serves as a solid forecasting base.

INTRODUCTION

Solving the timetabling problem involves arranging a university course timetable for all lectures in a week for a specific programme, where each lecture is assigned to a classroom and a time-slot. Students in indivisible groups who attend lectures at the same time are classified as homogeneous units called "sections". Each lecture contains one or more sections, as well as one or more faculty members assigned to it. The scheduler's job is to make sure that the teachers are available for the lecture at the scheduled time and that there are no scheduling conflicts between the various allocated lectures.

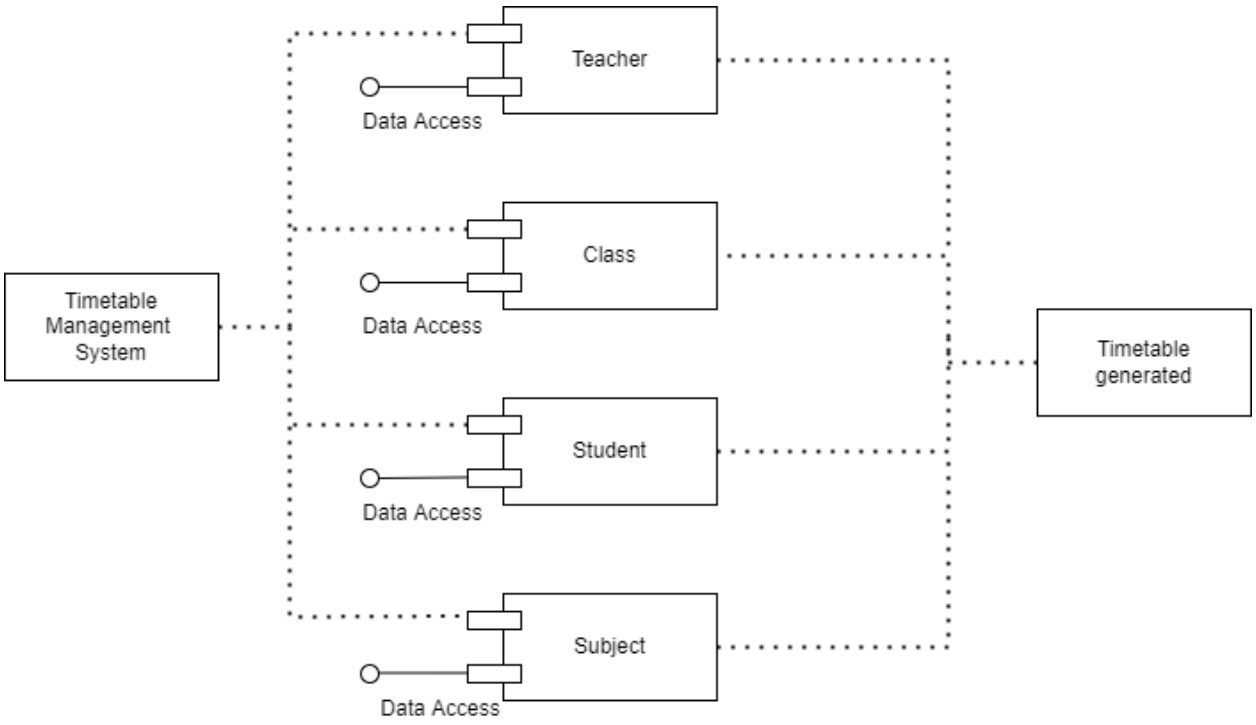
The essential need of the application is to submit information about the branch, topics, number of laboratories, and total number of periods. The tool then builds a timetable based on your specifications. It will aid in automating the time-consuming task of manually creating a timetable.

The proponents' way of tackling the timetabling problem comprises a Constraint Logic Programming approach implemented in a Constraint Logic Programming language such as MiniZinc.

EXISTING WORK

Postgraduates and undergraduates employ a manual timetable system in the current Time Table Management System. This manual system may result in data conflicts in the generated timetable. The current timetable was constantly changing, requiring numerous lecturers to handle it. It is consuming much too much of their time. The conclusion is that they require a system to assist and enhance the process of developing the timetable.

DATA FLOW DIAGRAM

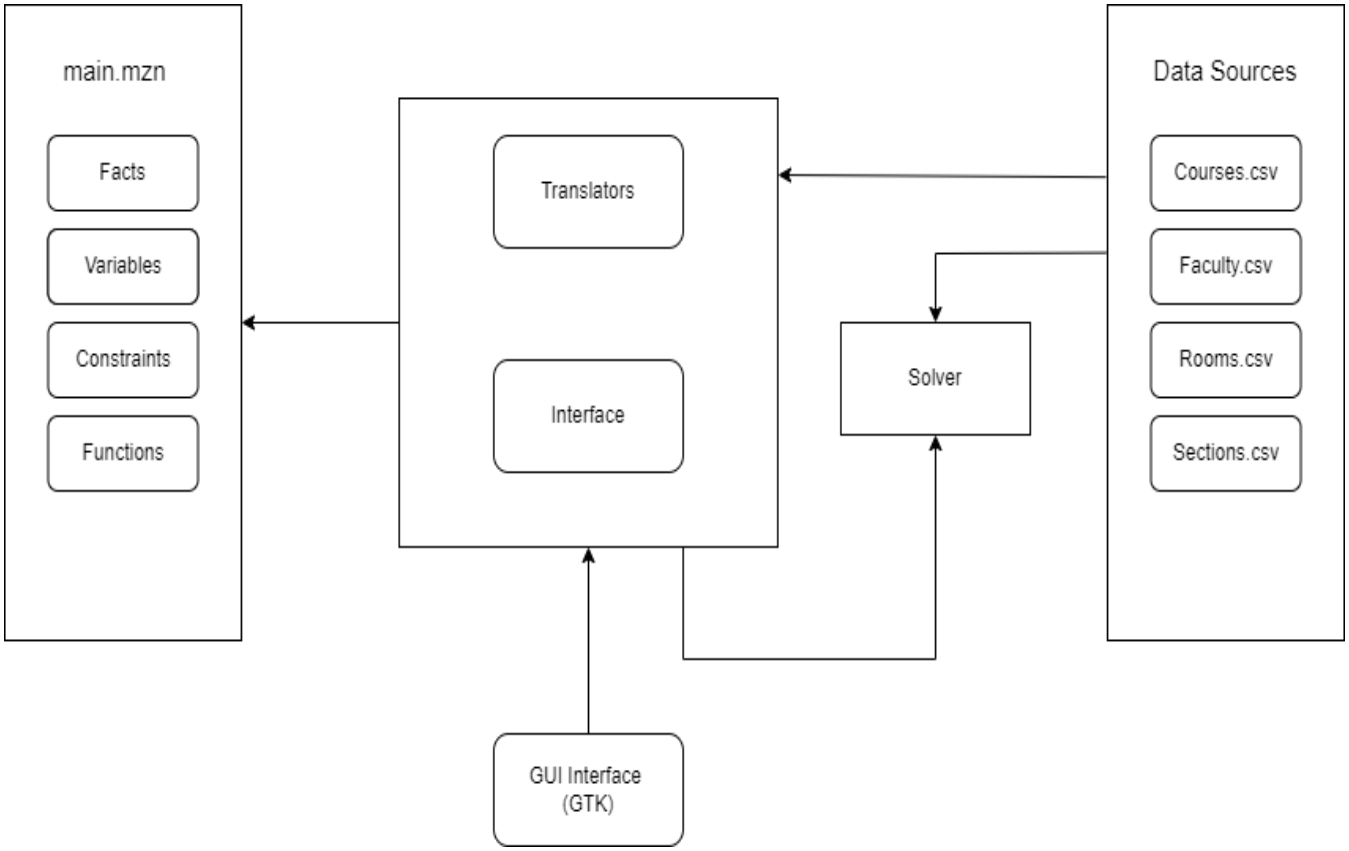


ARCHITECTURAL DIAGRAM

The application features a front-end built in C (using GTK) that will allow the user to generate a timetable after passing data and arguments to the MiniZinc solver. The solver runs the query and returns the results in a file stream, which can then be read by the C application and finally be displayed to the end user after formatting. The internals of the problem solver are hidden from the end user.

There are three main resources that need to be considered by the MiniZinc solver in the timetabling problem - the sections, the faculty and the classroom. Professors and Sections are already assigned to each other, and the system's only purpose is to assign a classroom to a particular section and professor pair for a particular time-slot. The solver ensures that none of the hard constraints are broken and also maximizes the number of soft constraints satisfied.

Predetermined data (such as available faculty and assigned sections) can be fed to the system by means of a .dzn file, where the data is formatted in a similar manner to defining variables in MiniZinc, or by directly reading a suitable data file (such as CSV).



ALGORITHM/CODE SAMPLES:

A) Coherence Constraints :

Module 1 :

INDIVIDUALITY CONSTRAINT : Faculty and Sections are only given courses assigned to them.

Input : Class list, section list, teacher list, number of slots per day,

```
constraint
forall (courseid in coursesarray)(
   forall(facultyindex in faculty_index_arr)(
       if sum(class in classes_per_week_for_single_fac_index_arr)
       (
           if facultydetails[facultyindex,class,3] == courseid
           then 1
           else 0
           endif
       ) < 1
       then forall(day in daysPerWeek, slot in slotsPerDay)(
           if facultyTimetable[facultyindex,day,slot] != 0
           then facultyTimetable[facultyindex,day,slot] != courseid
           endif
       )
       endif
   )
);
```

```
constraint
forall(sectionindex in sections_index_arr, day in daysPerWeek, slot in slotsPerDay)(
   if sectionTimetable[sectionindex, day, slot] > 0
   then sum(facultyindex in faculty_index_arr, class in
```

```
classes_per_week_for_single_fac_index_arr)(

       if facultydetails[facultyindex, class, 3] == sectionTimetable[sectionindex,
day, slot] /\

           facultydetails[facultyindex, class, 2] == sectionids_array[sectionindex]
/\

           facultyTimetable[facultyindex, day, slot] ==
sectionTimetable[sectionindex, day, slot]

       then 1

       else 0

       endif

   ) > 0

   endif

);
```

Module 2 :

ROOM-LIMITATION CONSTRAINT : A classroom is assigned just one lecture at a time. The number of teachers in the array of teachers that are teaching during any given slot must be lesser than or equal to the total number of rooms.

Input : Class list, section list, number of slots per day

```
constraint

forall(day in daysPerWeek, slot in slotsPerDay)(

   sum(facultyindex in faculty_index_arr)(

       if facultyTimetable[facultyindex,day,slot]!=0

       then 1

       else 0

       endif
```

```
   ) <= num_rooms

);
```

Module 3 :

COURSE-ENTITY CONSTRAINT : The sections and teachers involved in a particular class must share the same time slot. Two constraints are used to implement this. The first constraint ensures that if a teacher has a class scheduled on a particular day and slot, one of the student sections also has a class scheduled at that time. The second constraint ensures that if a student section has a class scheduled on a particular day and slot, a teacher also has a class scheduled at that time.

Input : Class list, section list, teachers list, number of slots per day

```
constraint
forall(facultyindex in faculty_index_arr,class in
classes_per_week_for_single_fac_index_arr)(
   forall(sec in sections_index_arr,day in daysPerWeek, slot in slotsPerDay)(
       if facultydetails[facultyindex,class,2] == sectionids_array[sec] /\
facultyTimetable[facultyindex,day,slot]!=0
       then sectionTimetable[sec,day,slot] = facultyTimetable[facultyindex,day,slot]
       endif
   )
);
```

B) Feature Constraints :

Module 4 :

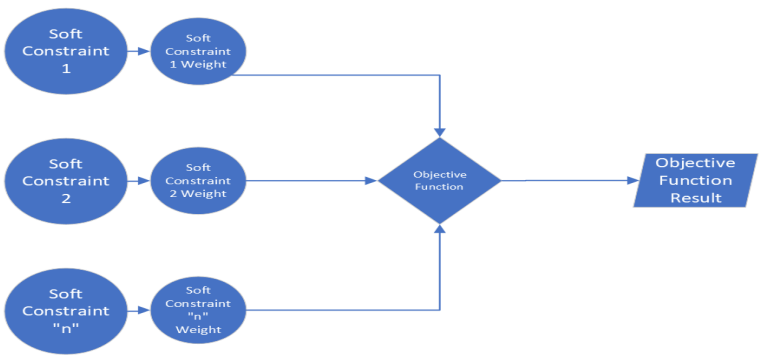
NON-REPEATING LECTURE CONSTRAINT :

The lecture of the same course should not be held multiple times on the same day. This feature can be built into the “Individuality Constraint”. The number of classes of a particular course on a particular day should be limited to one. The constraint below ensures that faculty do not have consecutive occupied slots unless the slot is occupied by a lab session, which may cover multiple slots.

```
constraint
```

```
forall(facultyindex in faculty_index_arr,day in daysPerWeek)(

   % Ensure faculty have free slot after every occupied slot unless

   % it is for a lab session.

   forall(slot in slotsPerDay)(

       if slot < slots

       then

           if facultyTimetable[facultyindex,day,slot] > 0 /\ not
courseid_is_lab(facultyTimetable[facultyindex,day,slot])

           then facultyTimetable[facultyindex,day,slot+1] = 0

           endif

       endif /\

       if slot > 1

       then

           if facultyTimetable[facultyindex,day,slot] > 0 /\ not
courseid_is_lab(facultyTimetable[facultyindex,day,slot])

           then facultyTimetable[facultyindex,day,slot-1] = 0

           endif

       endif

   )

);
```

C) General Soft Constraint Implementation



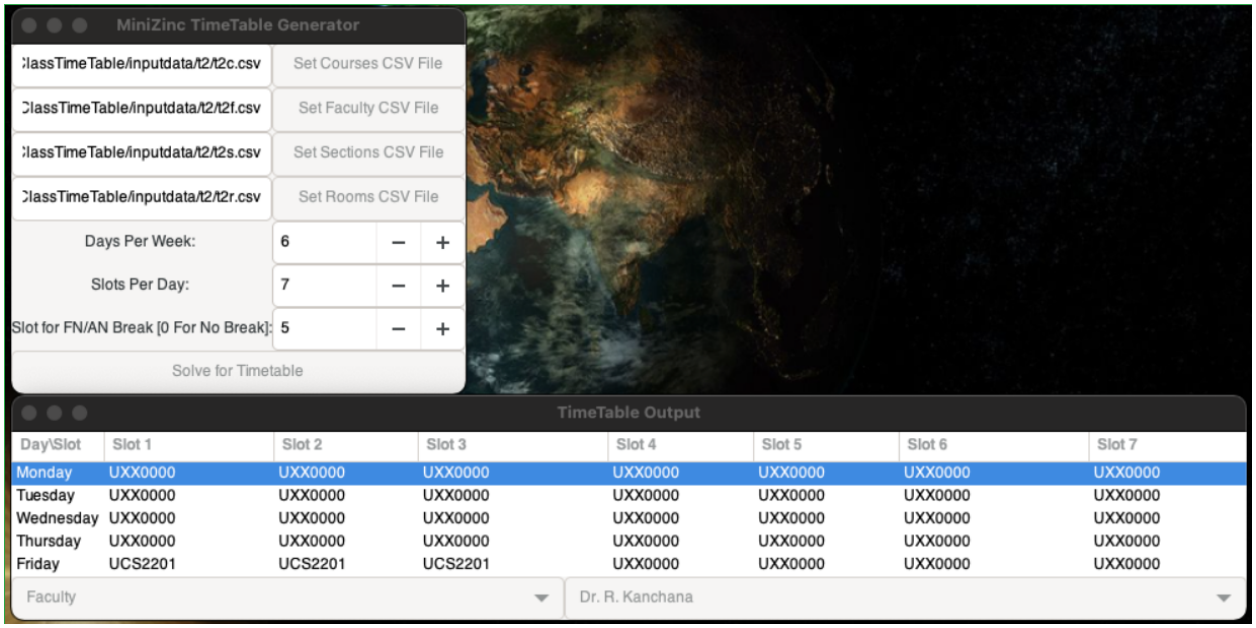
In order to implement recognition of soft constraints, an Objective Function can be used. Such a function would allocate equal or

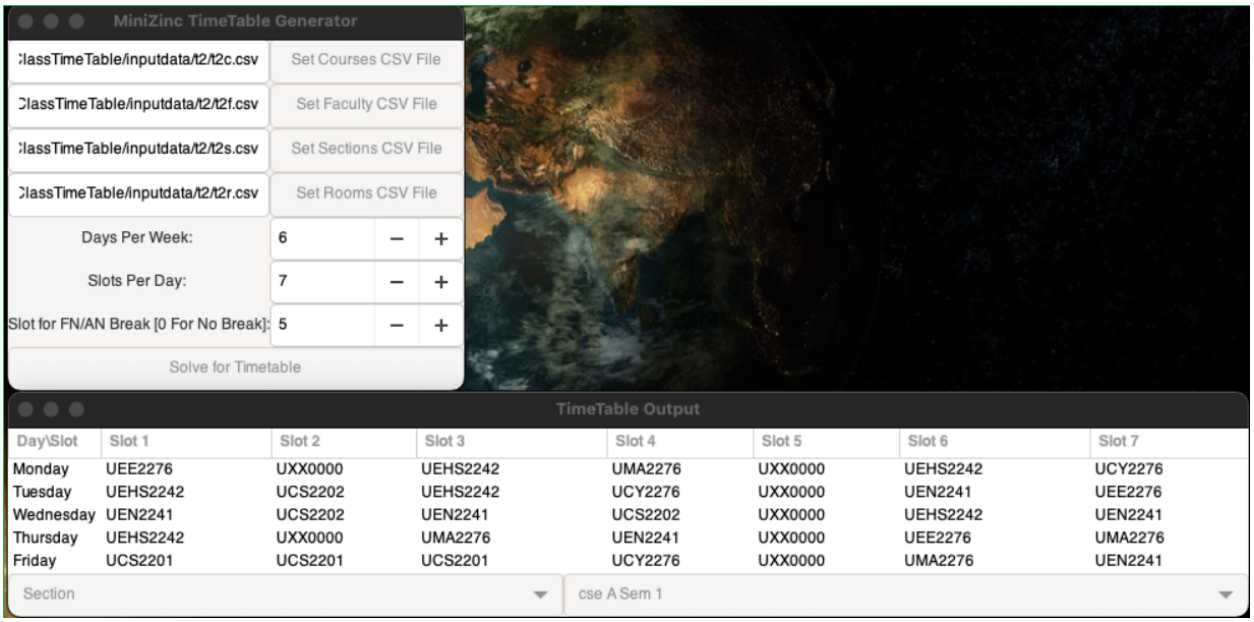
different weights to each soft constraint specified, and the solver would attempt to maximise the total return value of the objective function. The solution with the greatest objective function return value would then be returned as the ideal solution. The user may also choose which among the preference/soft constraints are included for the objective function. This choice is represented as a list of integers, representing both the priority of the soft constraint that the solver must use in order to calculate the return value of the objective function.

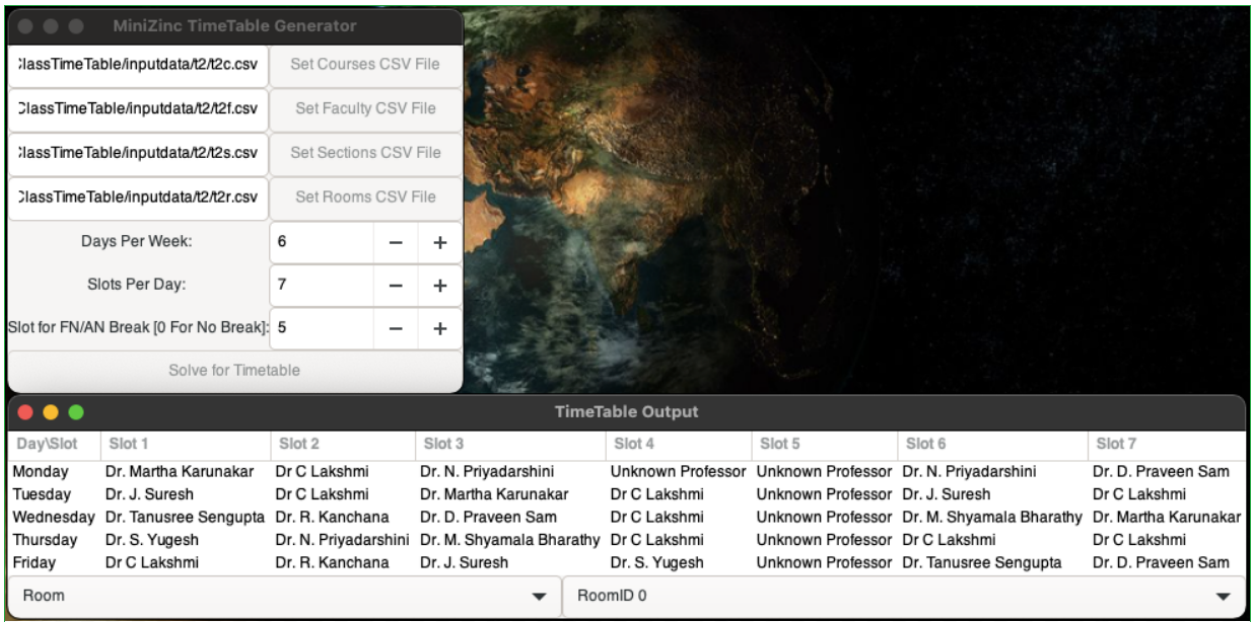
Further real-time analysis must be done in order to determine the weights that must be given to each soft constraint. It may also be necessary for the end user to be able to modify these weights as they require, or at least re-order the priority of the constraints, as each end user may have different soft constraint priorities in mind. For example, one user may find it more important to have classes alternate between Core and non-Core subjects, while another user may find it more important to avoid allocating core subjects in the last slot. To account for these differences, the end user must be allowed to edit the priority (if not the weight) of every soft constraint that can be taken into account.

TEST CASES

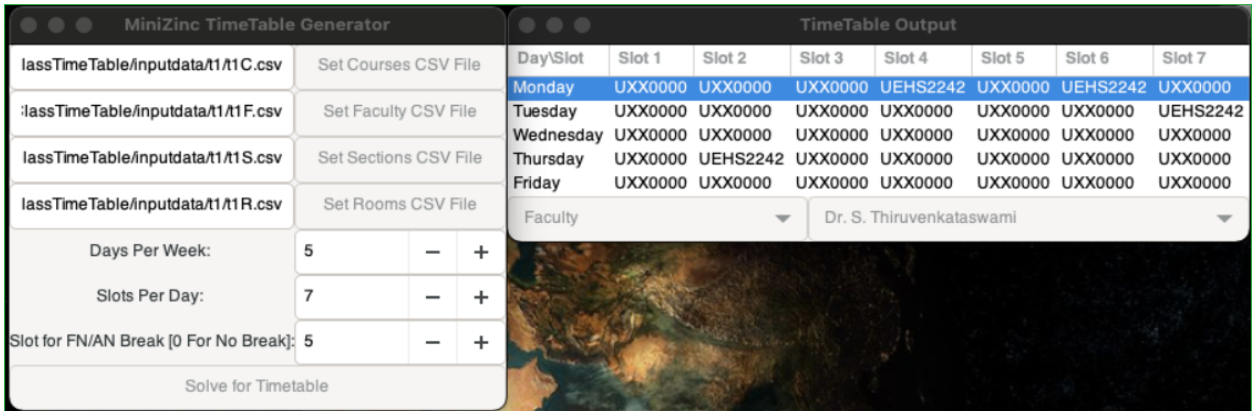
T1) A faculty can have maximum of 2 theory hours or one lab session per day.

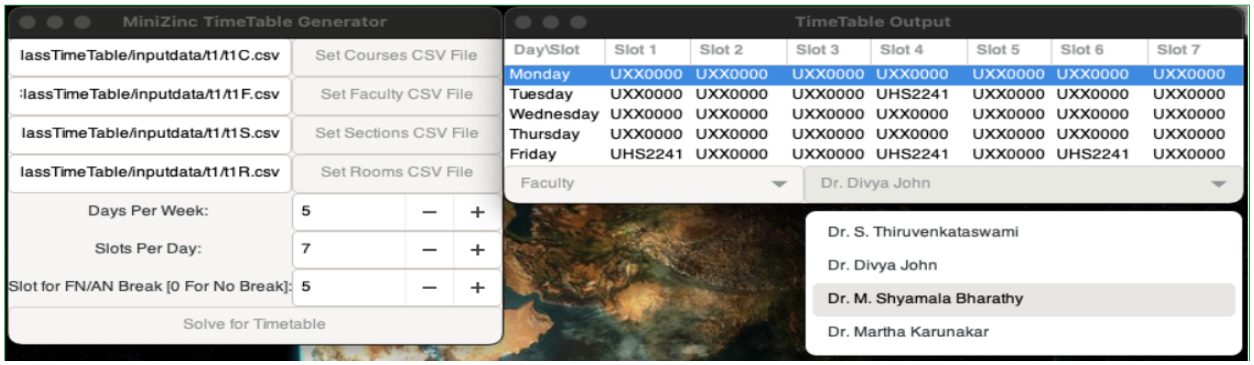


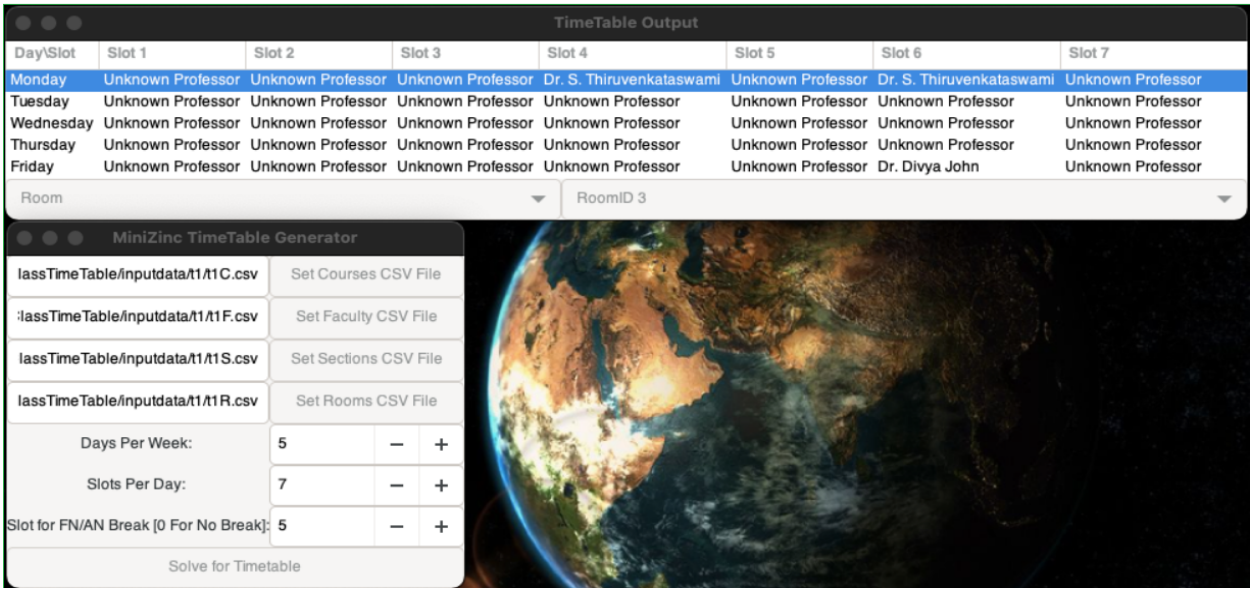


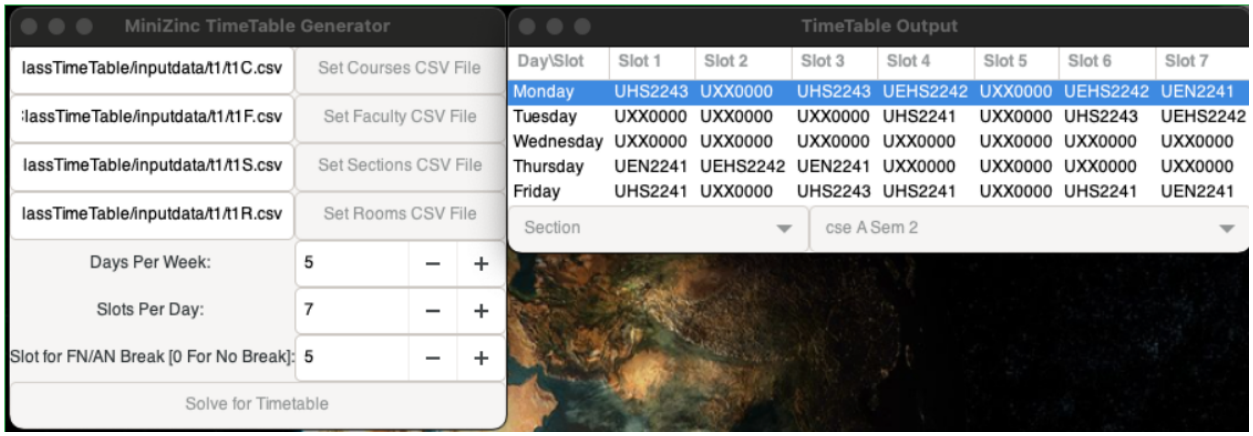


T2) Electives are scheduled at the same time across departments


MiniZinc TimeTable Generator
lassTimeTable/inputdata/t1/t1C.csv
Set Courses CSV File
:lassTimeTable/inputdata/t1/t1F.csv
Set Faculty CSV File
lassTimeTable/inputdata/t1/t1S.csv
Set Sections CSV File
lassTimeTable/inputdata/t1/t1R.csv
Set Rooms CSV File
Days Per Week: 5
Slots Per Day: 7
Slot for FN/AN Break [0 For No Break]: 5
Solve for Timetable
TimeTable Output
Day\Slot | Slot 1 | Slot 2 | Slot 3 | Slot 4 | Slot 5 | Slot 6 | Slot 7
Monday | UXX0000 | UXX0000 | UXX0000 | UEHS2242 | UXX0000 | UEHS2242 | UXX0000
Tuesday | UXX0000 | UXX0000 | UXX0000 | UXX0000 | UXX0000 | UXX0000 | UEHS2242
Wednesday | UXX0000 | UXX0000 | UXX0000 | UXX0000 | UXX0000 | UXX0000 | UXX0000
Thursday | UXX0000 | UEHS2242 | UXX0000 | UXX0000 | UXX0000 | UXX0000 | UXX0000
Friday | UXX0000 | UXX0000 | UXX0000 | UXX0000 | UXX0000 | UXX0000 | UXX0000
Faculty
Dr. S. Thiruvenkataswami


MiniZinc TimeTable Generator
lassTimeTable/inputdata/t1/t1C.csv
Set Courses CSV File
:lassTimeTable/inputdata/t1/t1F.csv
Set Faculty CSV File
lassTimeTable/inputdata/t1/t1S.csv
Set Sections CSV File
lassTimeTable/inputdata/t1/t1R.csv
Set Rooms CSV File
Days Per Week: 5
Slots Per Day: 7
Slot for FN/AN Break [0 For No Break]: 5
Solve for Timetable
TimeTable Output
Day\Slot | Slot 1 | Slot 2 | Slot 3 | Slot 4 | Slot 5 | Slot 6 | Slot 7
Monday | UXX0000 | UXX0000 | UXX0000 | UXX0000 | UXX0000 | UXX0000 | UXX0000
Tuesday | UXX0000 | UXX0000 | UXX0000 | UHS2241 | UXX0000 | UXX0000 | UXX0000
Wednesday | UXX0000 | UXX0000 | UXX0000 | UXX0000 | UXX0000 | UXX0000 | UXX0000
Thursday | UXX0000 | UXX0000 | UXX0000 | UXX0000 | UXX0000 | UXX0000 | UXX0000
Friday | UHS2241 | UXX0000 | UXX0000 | UHS2241 | UXX0000 | UHS2241 | UXX0000
Faculty
Dr. Divya John
Dr. S. Thiruvenkataswami
Dr. Divya John
Dr. M. Shyamala Bharathy
Dr. Martha Karunakar

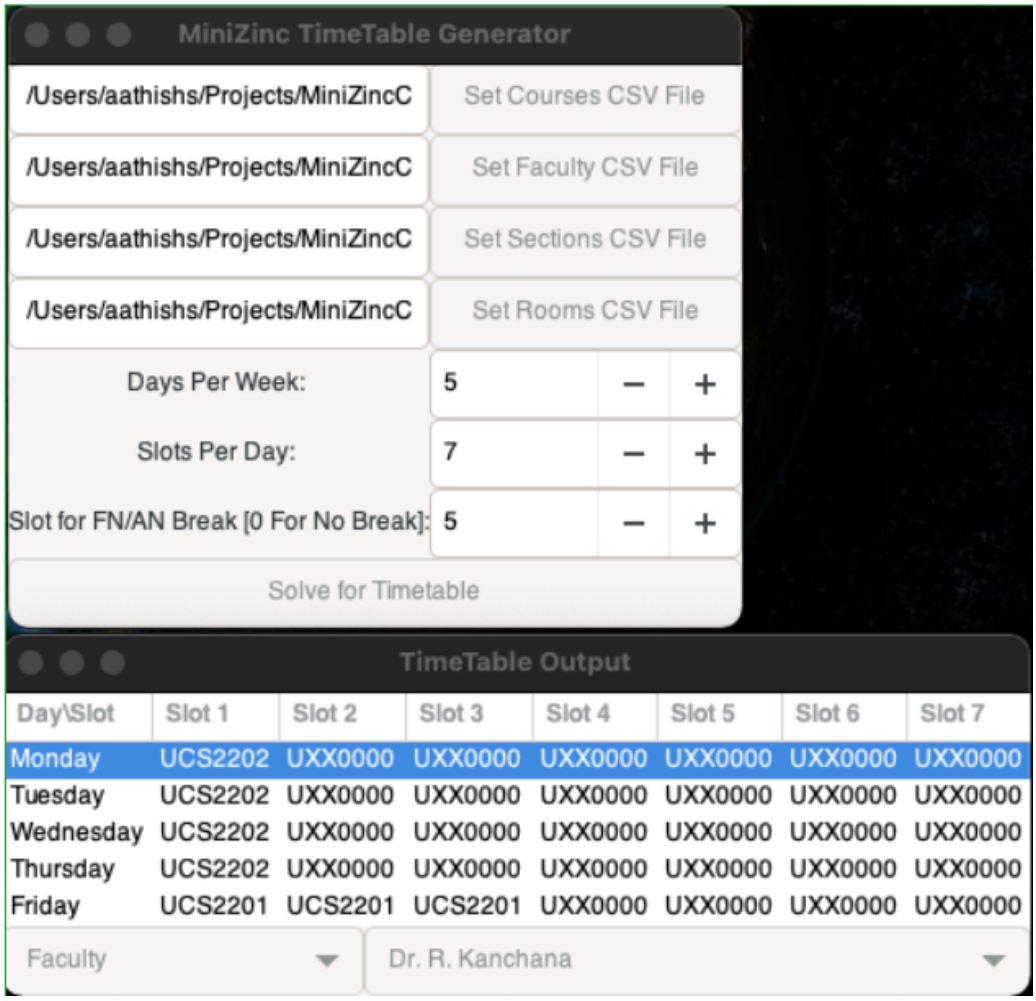

TimeTable Output
Day\Slot | Slot 1 | Slot 2 | Slot 3 | Slot 4 | Slot 5 | Slot 6 | Slot 7
Monday | Unknown Professor | Unknown Professor | Unknown Professor | Dr. S. Thiruvenkataswami | Unknown Professor | Dr. S. Thiruvenkataswami | Unknown Professor
Tuesday | Unknown Professor | Unknown Professor | Unknown Professor | Unknown Professor | Unknown Professor | Unknown Professor | Unknown Professor
Wednesday | Unknown Professor | Unknown Professor | Unknown Professor | Unknown Professor | Unknown Professor | Unknown Professor | Unknown Professor
Thursday | Unknown Professor | Unknown Professor | Unknown Professor | Unknown Professor | Unknown Professor | Unknown Professor | Unknown Professor
Friday | Unknown Professor | Unknown Professor | Unknown Professor | Unknown Professor | Unknown Professor | Dr. Divya John | Unknown Professor
Room
RoomID 3
MiniZinc TimeTable Generator
lassTimeTable/inputdata/t1/t1C.csv
Set Courses CSV File
:lassTimeTable/inputdata/t1/t1F.csv
Set Faculty CSV File
lassTimeTable/inputdata/t1/t1S.csv
Set Sections CSV File
lassTimeTable/inputdata/t1/t1R.csv
Set Rooms CSV File
Days Per Week: 5
Slots Per Day: 7
Slot for FN/AN Break [0 For No Break]: 5
Solve for Timetable

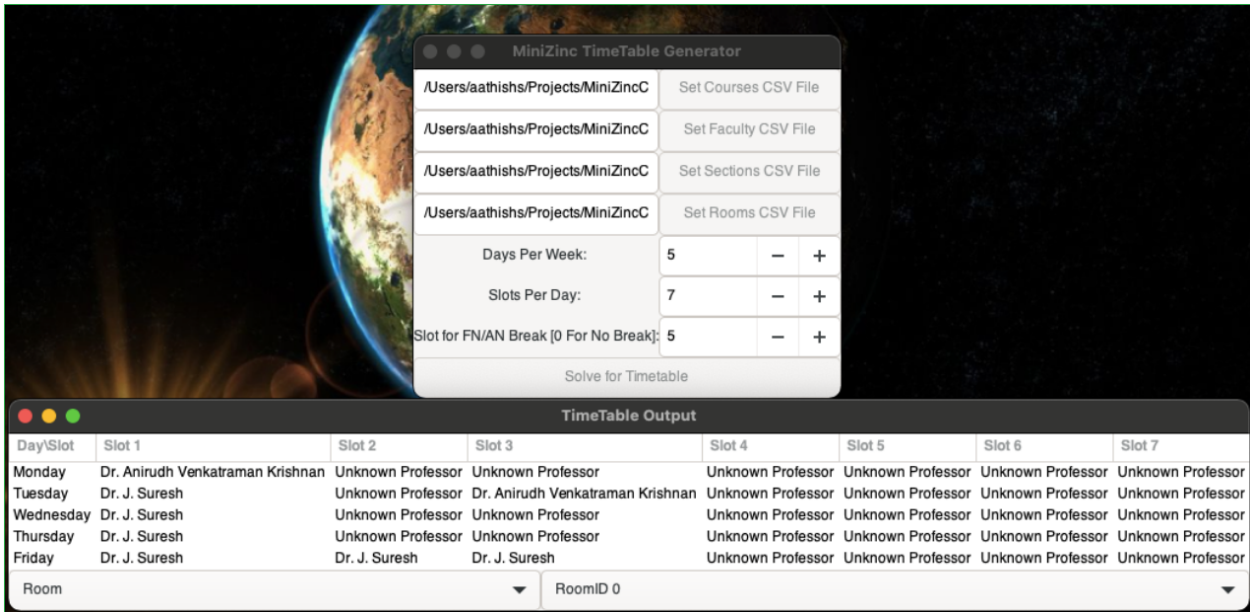


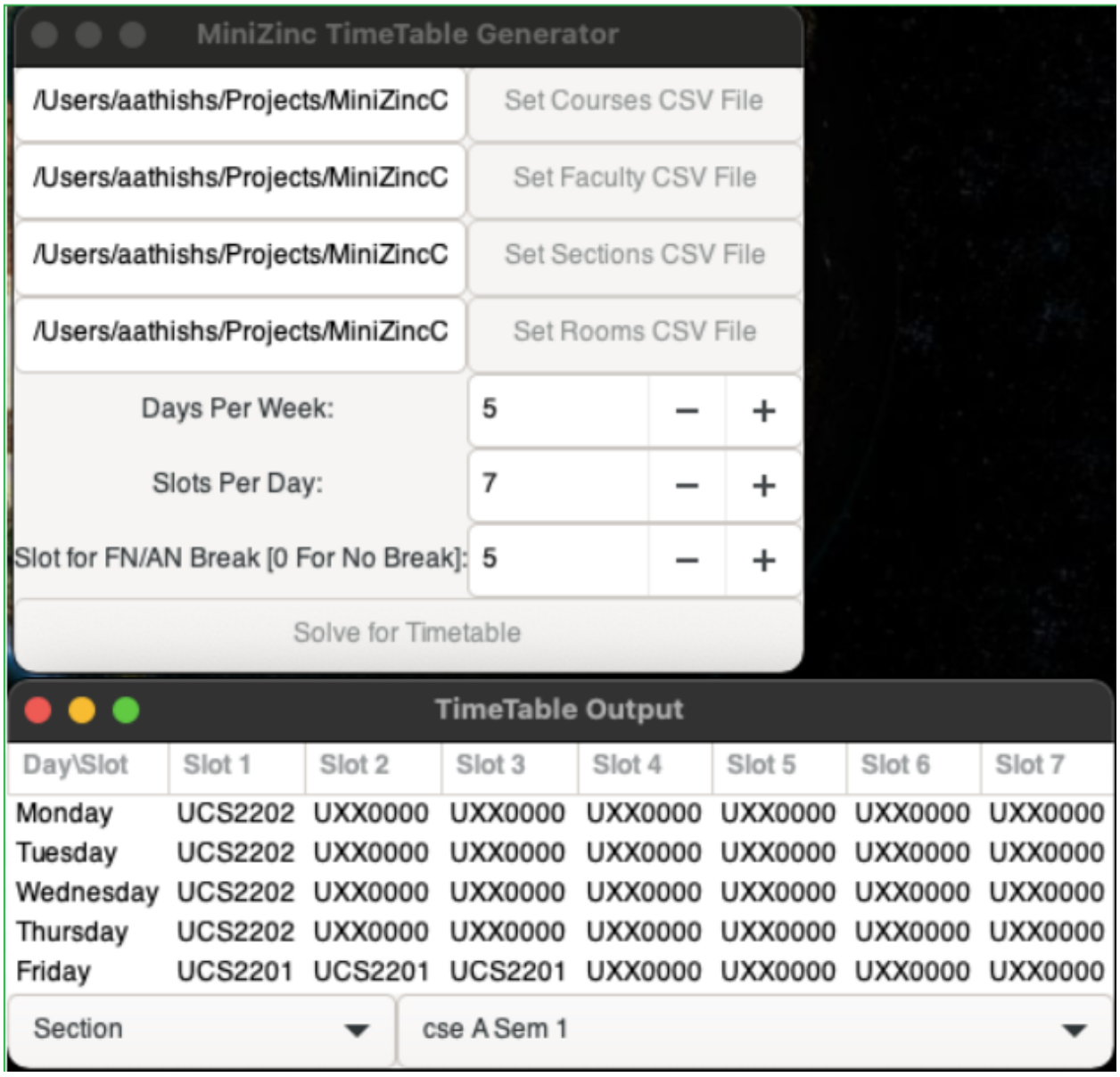
T3) Faculty are not allotted consecutive teaching hours.

Faculty and students are not allotted different classes at the same time.

Allotting core subjects in the last slot must be avoided.

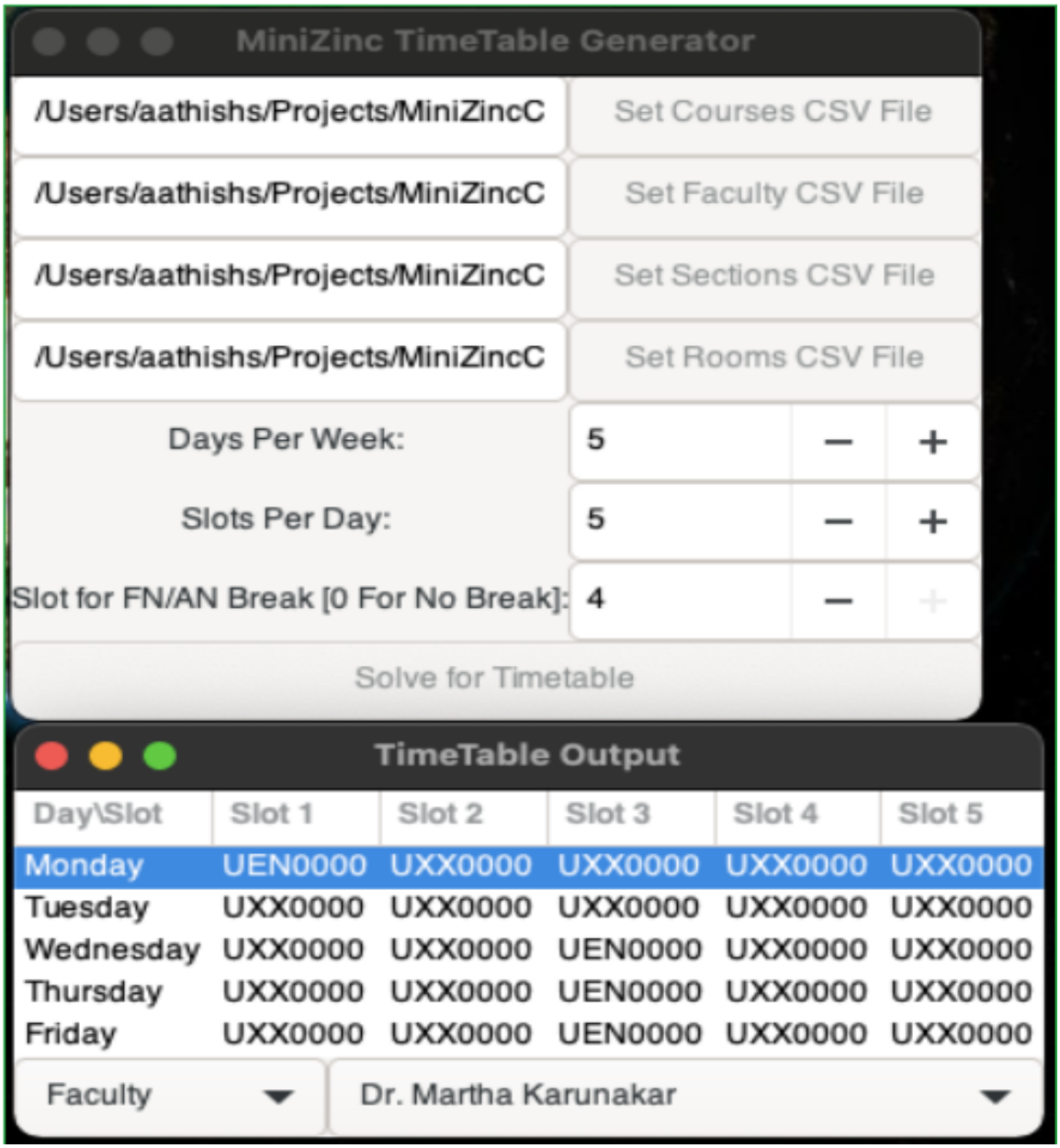

MiniZinc TimeTable Generator
/Users/aathishs/Projects/MiniZincC
Set Courses CSV File
/Users/aathishs/Projects/MiniZincC
Set Faculty CSV File
/Users/aathishs/Projects/MiniZincC
Set Sections CSV File
/Users/aathishs/Projects/MiniZincC
Set Rooms CSV File
Days Per Week: 5
Slots Per Day: 7
Slot for FN/AN Break [0 For No Break]: 5
Solve for Timetable
TimeTable Output
Day\Slot | Slot 1 | Slot 2 | Slot 3 | Slot 4 | Slot 5 | Slot 6 | Slot 7
Monday | UCS2202 | UXX0000 | UXX0000 | UXX0000 | UXX0000 | UXX0000 | UXX0000
Tuesday | UCS2202 | UXX0000 | UXX0000 | UXX0000 | UXX0000 | UXX0000 | UXX0000
Wednesday | UCS2202 | UXX0000 | UXX0000 | UXX0000 | UXX0000 | UXX0000 | UXX0000
Thursday | UCS2202 | UXX0000 | UXX0000 | UXX0000 | UXX0000 | UXX0000 | UXX0000
Friday | UCS2201 | UCS2201 | UCS2201 | UXX0000 | UXX0000 | UXX0000 | UXX0000
Faculty
Dr. R. Kanchana

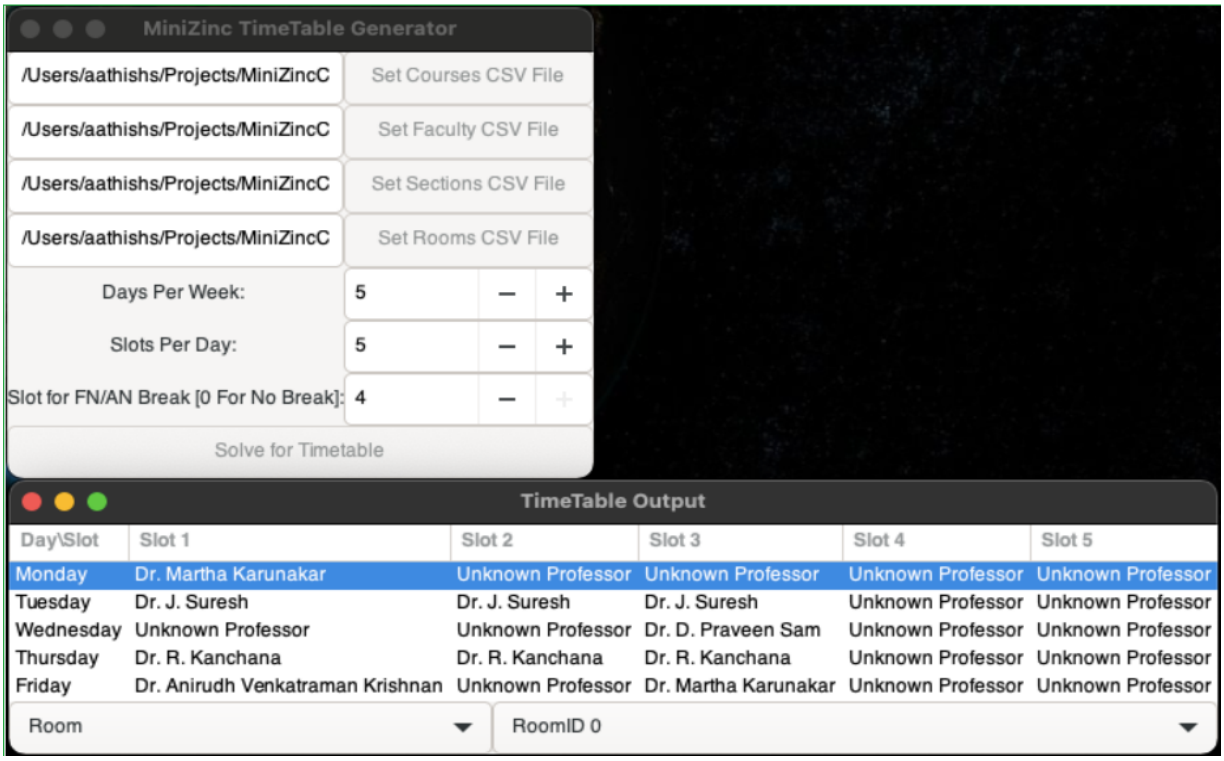

MiniZinc TimeTable Generator
/Users/aathishs/Projects/MiniZincC
Set Courses CSV File
/Users/aathishs/Projects/MiniZincC
Set Faculty CSV File
/Users/aathishs/Projects/MiniZincC
Set Sections CSV File
/Users/aathishs/Projects/MiniZincC
Set Rooms CSV File
Days Per Week: 5
Slots Per Day: 7
Slot for FN/AN Break [0 For No Break]: 5
Solve for Timetable
TimeTable Output
Day\Slot | Slot 1 | Slot 2 | Slot 3 | Slot 4 | Slot 5 | Slot 6 | Slot 7
Monday | Dr. Anirudh Venkatraman Krishnan | Unknown Professor | Unknown Professor | Unknown Professor | Unknown Professor | Unknown Professor | Unknown Professor
Tuesday | Dr. J. Suresh | Unknown Professor | Dr. Anirudh Venkatraman Krishnan | Unknown Professor | Unknown Professor | Unknown Professor | Unknown Professor
Wednesday | Dr. J. Suresh | Unknown Professor | Unknown Professor | Unknown Professor | Unknown Professor | Unknown Professor | Unknown Professor
Thursday | Dr. J. Suresh | Unknown Professor | Unknown Professor | Unknown Professor | Unknown Professor | Unknown Professor | Unknown Professor
Friday | Dr. J. Suresh | Dr. J. Suresh | Dr. J. Suresh | Unknown Professor | Unknown Professor | Unknown Professor | Unknown Professor
Room
RoomID 0

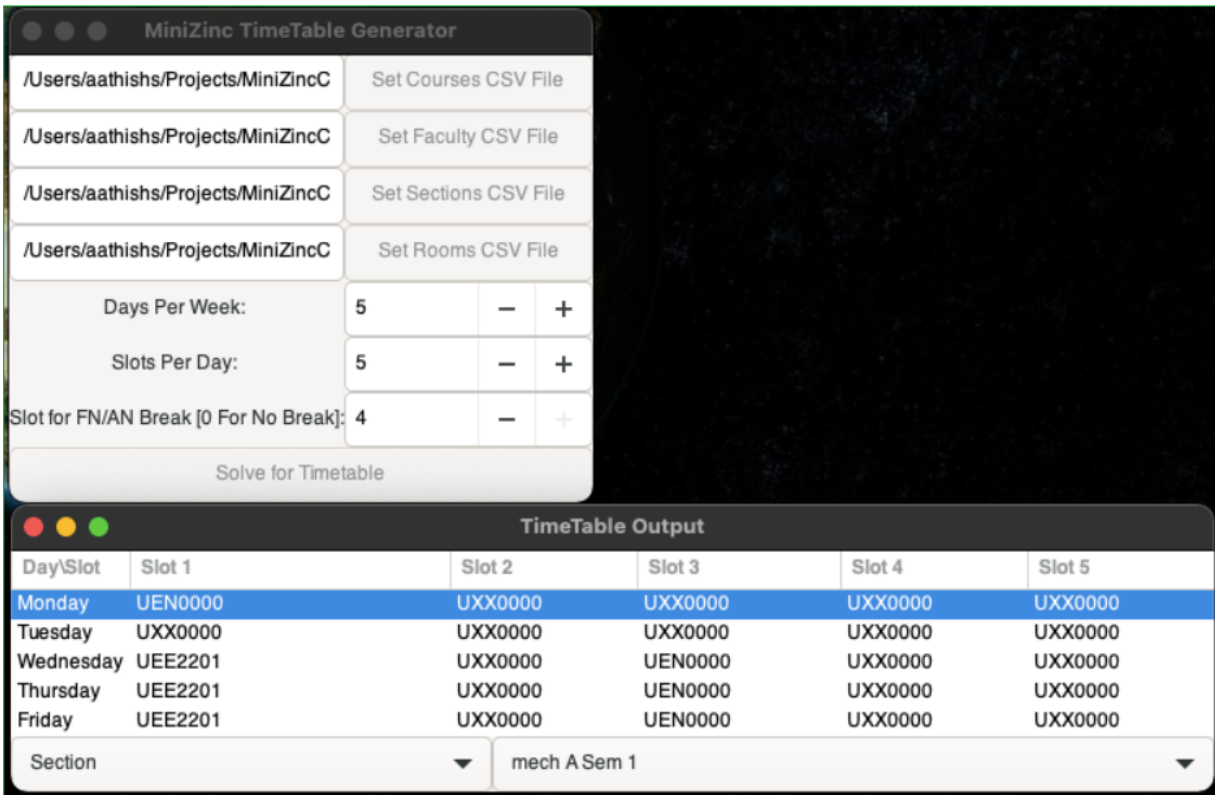

MiniZinc TimeTable Generator
/Users/aathishs/Projects/MiniZincC
Set Courses CSV File
/Users/aathishs/Projects/MiniZincC
Set Faculty CSV File
/Users/aathishs/Projects/MiniZincC
Set Sections CSV File
/Users/aathishs/Projects/MiniZincC
Set Rooms CSV File
Days Per Week:
5
Slots Per Day:
7
Slot for FN/AN Break [0 For No Break]:
5
Solve for Timetable
TimeTable Output
Day\Slot
Slot 1
Slot 2
Slot 3
Slot 4
Slot 5
Slot 6
Slot 7
Monday UCS2202 UXX0000 UXX0000 UXX0000 UXX0000 UXX0000 UXX0000
Tuesday UCS2202 UXX0000 UXX0000 UXX0000 UXX0000 UXX0000 UXX0000
Wednesday UCS2202 UXX0000 UXX0000 UXX0000 UXX0000 UXX0000 UXX0000
Thursday UCS2202 UXX0000 UXX0000 UXX0000 UXX0000 UXX0000 UXX0000
Friday UCS2201 UCS2201 UCS2201 UXX0000 UXX0000 UXX0000 UXX0000
Section
cse A Sem 1

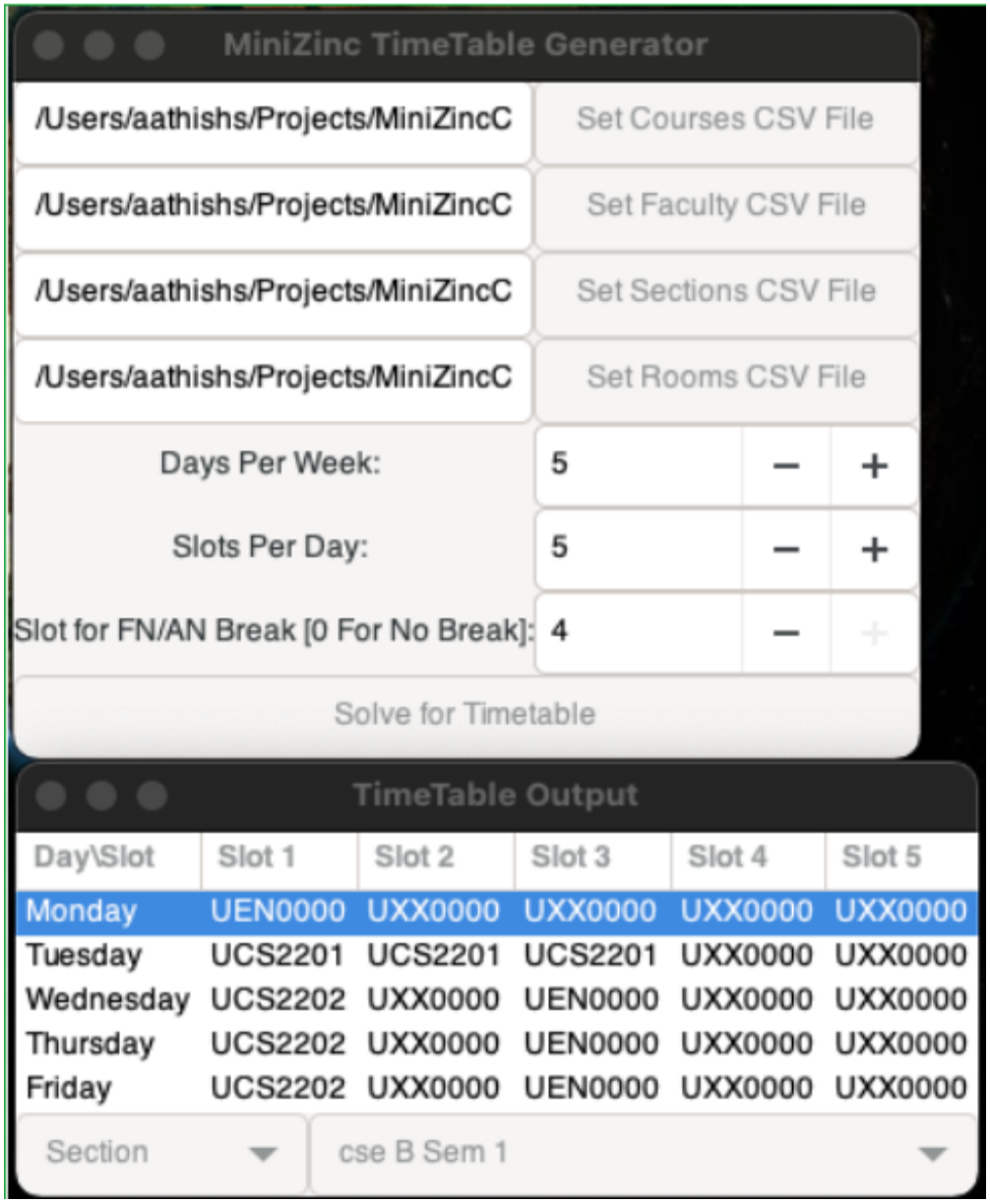
T4) A classroom is just assigned one lecture at a time.

Sessions alternate between core and non-core subjects.




MiniZinc TimeTable Generator
/Users/aathishs/Projects/MiniZincC
Set Courses CSV File
/Users/aathishs/Projects/MiniZincC
Set Faculty CSV File
/Users/aathishs/Projects/MiniZincC
Set Sections CSV File
/Users/aathishs/Projects/MiniZincC
Set Rooms CSV File
Days Per Week: 5
Slots Per Day: 5
Slot for FN/AN Break [0 For No Break]: 4
Solve for Timetable
TimeTable Output
Day\Slot | Slot 1 | Slot 2 | Slot 3 | Slot 4 | Slot 5
Monday | Dr. Martha Karunakar | Unknown Professor | Unknown Professor | Unknown Professor | Unknown Professor
Tuesday | Dr. J. Suresh | Dr. J. Suresh | Dr. J. Suresh | Unknown Professor | Unknown Professor
Wednesday | Unknown Professor | Unknown Professor | Dr. D. Praveen Sam | Unknown Professor | Unknown Professor
Thursday | Dr. R. Kanchana | Dr. R. Kanchana | Dr. R. Kanchana | Unknown Professor | Unknown Professor
Friday | Dr. Anirudh Venkatraman Krishnan | Unknown Professor | Dr. Martha Karunakar | Unknown Professor | Unknown Professor
Room
RoomID 0


MiniZinc TimeTable Generator
/Users/aathishs/Projects/MiniZincC
Set Courses CSV File
/Users/aathishs/Projects/MiniZincC
Set Faculty CSV File
/Users/aathishs/Projects/MiniZincC
Set Sections CSV File
/Users/aathishs/Projects/MiniZincC
Set Rooms CSV File
Days Per Week: 5
Slots Per Day: 5
Slot for FN/AN Break [0 For No Break]: 4
Solve for Timetable
TimeTable Output
Day\Slot | Slot 1 | Slot 2 | Slot 3 | Slot 4 | Slot 5
Monday | UEN0000 | UXX0000 | UXX0000 | UXX0000 | UXX0000
Tuesday | UXX0000 | UXX0000 | UXX0000 | UXX0000 | UXX0000
Wednesday | UEE2201 | UXX0000 | UEN0000 | UXX0000 | UXX0000
Thursday | UEE2201 | UXX0000 | UEN0000 | UXX0000 | UXX0000
Friday | UEE2201 | UXX0000 | UEN0000 | UXX0000 | UXX0000
Section
mech A Sem 1


MiniZinc TimeTable Generator
/Users/aathishs/Projects/MiniZincC
Set Courses CSV File
/Users/aathishs/Projects/MiniZincC
Set Faculty CSV File
/Users/aathishs/Projects/MiniZincC
Set Sections CSV File
/Users/aathishs/Projects/MiniZincC
Set Rooms CSV File
Days Per Week:
5
Slots Per Day:
5
Slot for FN/AN Break [0 For No Break]:
4
Solve for Timetable
TimeTable Output
Day\Slot
Slot 1
Slot 2
Slot 3
Slot 4
Slot 5
Monday UEN0000 UXX0000 UXX0000 UXX0000 UXX0000
Tuesday UCS2201 UCS2201 UCS2201 UXX0000 UXX0000
Wednesday UCS2202 UXX0000 UEN0000 UXX0000 UXX0000
Thursday UCS2202 UXX0000 UEN0000 UXX0000 UXX0000
Friday UCS2202 UXX0000 UEN0000 UXX0000 UXX0000
Section
cse B Sem 1

CONCLUSION

For time table administration, a web application was created. It is a mechanism designed to give improved support to college employees. The schedule module serves as a solid forecasting base. The time table management system includes the ability to create and examine timetables for particular employees and students.

REFERENCES

1. [2011.07507] Automated Large-scale Class Scheduling in MiniZinc
2. University Course Timetabling with Soft Constraints
3. https://www.minizinc.org