# P1: Timetable Management System for an Academic Institution

Project Review - 4

Team 7

Design
B4
Coding

Team Members:
Aathish Sivasubrahmanian (3122 21 5001 001)
Ayshwarya Badhrinath (3122 21 5001 017)
Harshida Sujatha Palaniraj (3122 21 5001 031)

# Problem Statement

**Generate a University Timetable scheduling periods for the given subjects keeping in mind the various soft and hard constraints specified.**

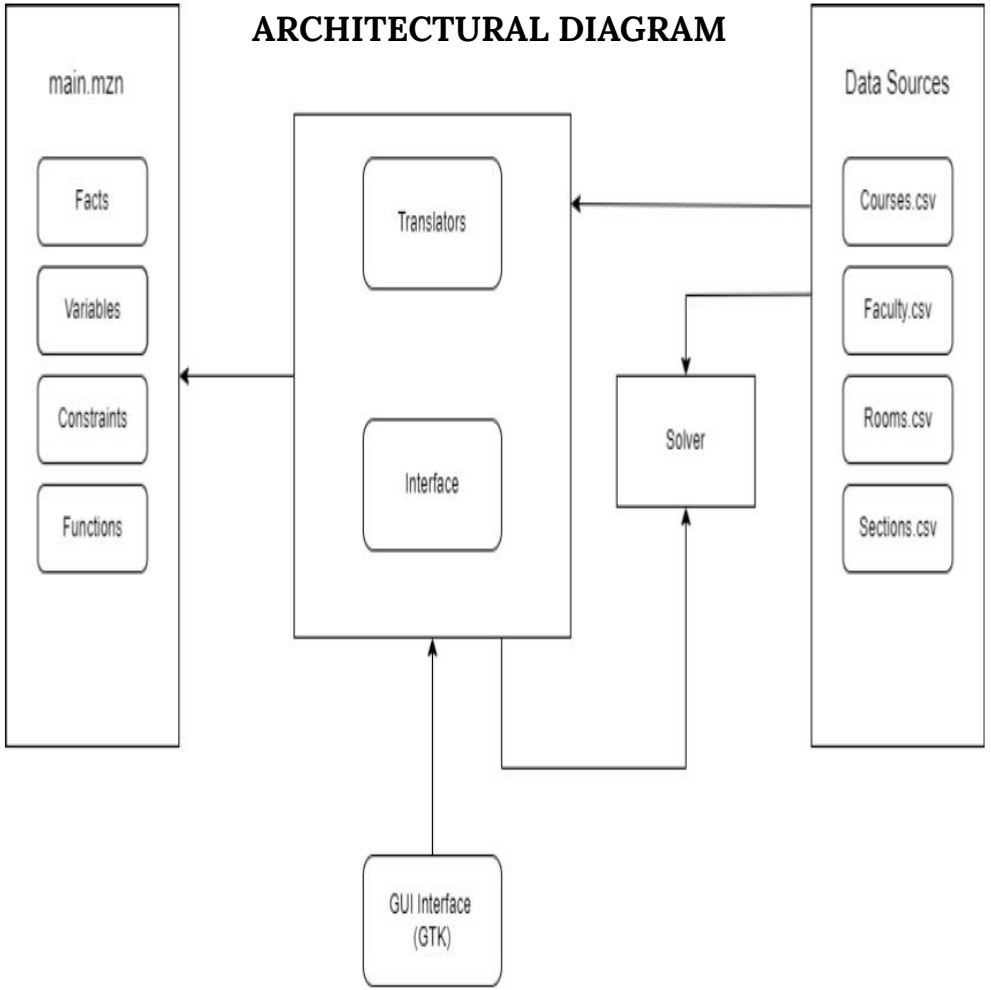
# Input & Output

## Inputs:

- General Information
  - Degree (UG/PG)
  - Semester
- Resources
  - Classrooms
  - Physical lab and slots
  - Number of hours per course.
  - Faculty available
  - Students undertaking a course grouped into sections.

## Outputs:

- Class Timetable for each section.
- Faculty Timetable for each faculty member.
- Lab timetable
- Room timetable



## TECHNIQUE - CONSTRAINT LOGICAL PROGRAMMING (CLP)

With the aid of an universal constraint-propagation engine and a declarative problem description, constraint programming is a technique for resolving highly combinatorial problems.

MiniZinc, a high-level modeling language, where declaring relevant variables and parameters and their relations in the form of constraints, is enough to generate an “optimized” or “satisfiable” solution, with the help of a solver has been incorporated to solve the given problem.

The overall layout of a schedule programme in this Constraint Satisfaction Problem, include :

- Specifying the domains of the problem variable
- Stating the constraints
- Producing the values

In CLP, the time periods and rooms are represented in terms of integers inside given domains. Along with defining all the data-structures, the MiniZinc model defines the all possible relations among them. These relations are called constraints.
The constraints are defined independently in the MiniZinc IDE and are compiled together using the ‘solve-satisfy’ method (hard constraints). The soft constraints are compiled using the ‘solve-minimize' method.
I.e., In MiniZinc, the model does not need to explicitly declare the search algorithm to find a satisfiable or optimized solution.
Any FlatZinc supported solver can be integrated with the MiniZinc model to obtain the required solution.

In our case, we have implemented the Chuffed Solver. Chuffed is a lazy clause generation solver. It combines Finite Domain (FD) propagation with Boolean satisfiability.

# Soft Constraints

- Avoidance of allotment of Core subjects in the last time slot
- Allotment of Co-curricular / Non-credited subjects to the afternoon sessions.
- Alternate sessions between Core and non-Core subjects.
- Allotment of the First hour session to department faculty.
- Allotment of first hour to an accredited course.

# REFERENCES

[2011.07507] Automated Large-scale Class Scheduling in MiniZinc

University Course Timetabling with Soft Constraints

https://www.minizinc.org