Proposal

TimeTable Generator

Team Members:
Aman Yadav
DakshKumar Gondaliya
Kirtika Singhal
Mayank Pathela
Nikhil Sachan
Parmeshwar Kumawat

Abstract

The manual system of preparing timetable in schools with a large number of sections of classes is very time-consuming. So we are trying to develop a software which will reduce the difficulties and time required for generating a timetable by using constraint-based programming. The system will take various inputs like details of teachers, subjects and class sections, depending upon these inputs it will generate a possible timetable.

1. Introduction

Management of almost every school has to handle three types of resources: teachers, students, and rooms. The manual lecture timetable scheduling demands considerable time and efforts. It is a Constraint satisfaction problem where we find a solution that satisfies the given set of constraints like for this problem some of the constraints are: a teacher can’t have more than one lecture at a time slot and a class cannot have two lectures at the same slot. So we are designing a software that will take various inputs like details of subjects, teachers, and sections of a class along with the maximum workload for a faculty for a week to generate a possible timetable satisfying as much constraints as possible. The software we will develop will be a web application having different interfaces for admin and teachers.

1.1 Admin End

It is an interface for admin where he can generate the timetable and modify it. The algorithm will take 3-5 minutes to execute and give a possible solution.

1.1.1 Registration

When the admin visits the web application for the first time and wants to generate the timetable, he will be asked to register an account. This page will ask him to enter his username and password.

1.1.2 Admin Selection

This page will contain two options:

- Data entry page for generating a timetable
- View or Modify the generated timetable

1.1.3 Data Entry

This page will require the admin to input the list of teachers, subjects, class sections, and the maximum number of lectures for that subject in a week. After filling in these details he needs to submit it.

1.1.4 Mappings

This page will provide admin with a separate dropdown list of teachers, subjects, and sections from which he has to select and fill in a row that will be stored as a tuple in the database. There will be a button to add another such row and delete a row. The drop-down list will reduce the efforts of typing the name of a teacher again and again. Also there will be a button at the bottom of the page to generate the timetable.

1.1.5 Generated Timetable

This page will display the timetable generated and the admin can modify some of the entries in the timetable and submit them in order to update the database.

1.2 Teacher End

There is a separate interface for the teachers where they can only view the timetable.

1.2.1 Log In

All the teachers will be given a common username and password. They need to log in through that.

1.2.2 Selection

It is an interface that provides user to select a teacher whose timetable he wants to see from the dropdown list of teachers. This way he can easily see the classes scheduled for him.

1.2.3 View scheduled classes

It is an interface that will display the classes scheduled for the teacher.

2. Proposed Implementation Plans and Timeline

2.1 Project Timeline

The following is an abstract high-level Proposed Project Plan and Timeline:

1. The Project Duration is 2 to 2.5 months. The End of the project will be denoted by successful testing and small-scale Deployment.
2. The Survey Phase and Software Requirement Specifications are set to last for approximately 2 weeks.
3. After completion of requirement analysis, next comes Design Phase which is set to last around 1-2 weeks.
4. Next, comes the Coding and unit testing phase which is set to last for 3 to 4 weeks.
5. Finally if the time permits we will do small-scale deployment server.

2.2 Tools and Technology

Following is the list of tools and technologies we are planning to use(they may change if extra requirements arrive):

1. React Framework
2. Node.js
3. MongoDB
4. AWS cloud storage

3. Requirements and Assumptions

3.1 Requirements

We will require Amazon Web Service(AWS) for the database.

3.2 Assumptions

We have made certain assumptions regarding the timetable. Some of them are:

- The schools will have access to Computers and internet connections.
- Teachers are available for all the working days in a week, i.e, no preferences are taken into consideration.

4. Project Risks

There may occur a chance where the algorithm may not show any solution. In that case, we will start relaxing the constraints and generate a new timetable.

5. Project Risks

The major challenge in our project is to minimize the number of clashes, ie, a class should not have two lectures at the same time and a teacher can’t have more than one lecture at the same slot. We have to minimize the time of execution also.

6. Future Extensions

In the future, we can expand it for colleges for which we have to add extra constraints for the backlog, preference of visiting faculty.