

# **SOFTWARE DESIGN DOCUMENT**

## **Time-table generator**

Table of Contents	Page numbers
1. Introduction	2
1.1 Purpose	2
1.2 Scope	2
1.3 Definitions, Acronyms and Abbreviations	2
2. System Overview	2
3. System Components	3
3.1 Decomposition Description	3
3.2 Dependency Description	4
3.3 Interface Description	6
4. Detailed Design	7
4.1 RTM	7

### **TEAM MEMBERS:**

Akshay Prasad

PES1UG21CS065

Adithya Sharma C A

PES1UG21CS034

Ajey Bhat

PES1UG21CS053

Aditya Korishetty

PES1UG21CS914

# **1. Introduction**

## **1.1 Purpose**

The Software Design Document describes the architecture and system design for Time Table Generator, a tool that generates a time table with any conflicts.

Time Table Generator is designed to help Universities,schools . This document is intended for Project Managers, Software Engineers, and anyone else who will be involved in the implementation of the system.

## **1.2 Scope**

This document describes the implementation details of the Time Table Generator which will consist of four major components:file handling operations(Taking input), approval , time table generation, approval from HOD, Display to students. Each of the components will be explained in detail in this Software Design Document.

## **1.3 Definitions, Acronyms and Abbreviations**

- 1) CSV File: Comma separated value files
- 2) PDF: Portable document file
- 3) HOD: Head of department
- 4) SRS: Software requirement specification

## **2. System Overview**

The input manager passes the inputs in the form of a csv file to the time table generator. The input file consists of the sections, no. of slots ,teachers ,their respective subjects

We use various file handling operations to read this csv file and create a graph with each node representing the subject and the section.

Using a graph coloring algorithm, we create a timetable that doesn't have any conflicts

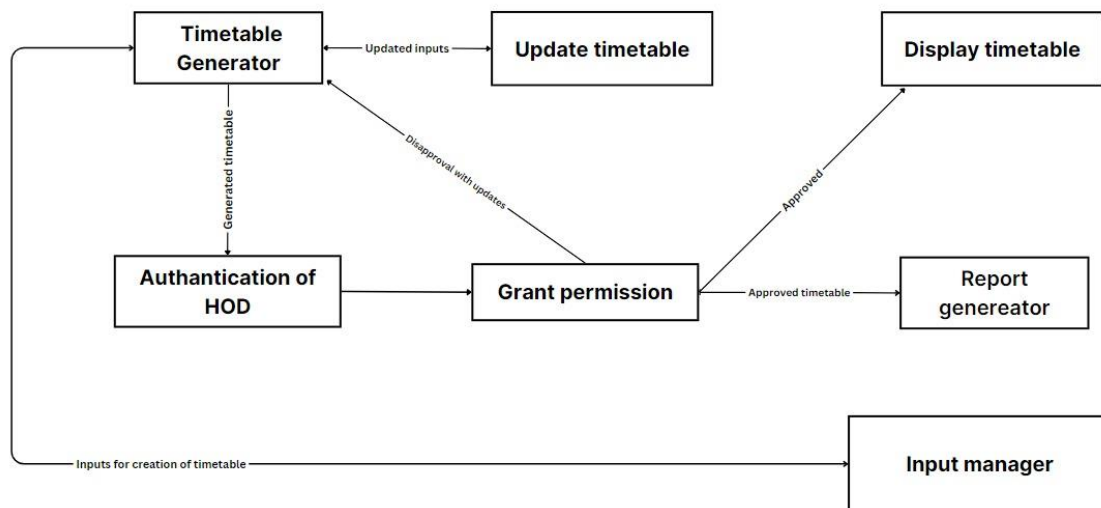
Once the graph is coloured, we create a time table by assigning nodes of the same color in the same time slot.

Now we populate a csv files and assign subjects to different time slots of various sections

This is then open to students to view

### 3. System Components

#### 3.1 Decomposition description



The main component in this project is the Timetable Generator.

There are 3 parts in this-

**i) File handling** ( Taking in input from csv file)

The file handling part includes reading various inputs from the 'input.csv' file such as teacher and their respective subjects, sections and storing in appropriate data structures.

**ii) Graph modeling** -

Graph modeling is using the data structures to create a graph

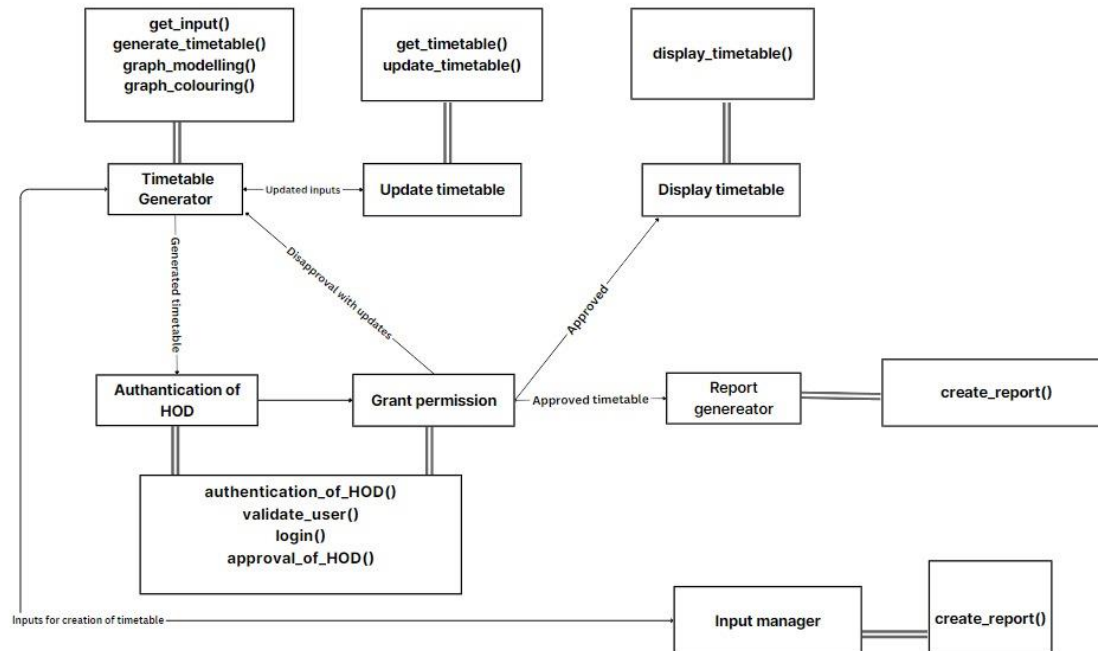
**iii) Graph coloring** - Using a graph coloring algorithm, we assign appropriate colors to nodes (section - subject) . Nodes having the same color can be present in the same time slot. Nodes having the same color CANNOT be present in the same time slot.

Once the graph is coloured, we create a time table by assigning nodes of the same color in the same time slot.

#### iv) Representation in csv format

Once the algorithm is applied we represent this in a csv file in order to enhance readability.

### 3.2 Dependency Description



The following major functionalities and their inputs and return values have been listed below.

#### 1) Input manager:

create\_report()

Input: Timetable in the form of CSV file.

Return: Report generated to be written onto a file.

#### 2) Time table generator

get\_input()

Input: Receives a CSV file

Return: Store data in the data structure and return the data structure.

generate\_timetable()

Input: Data-structure returned from get\_input()

Helper functions:

a) graph\_modelling()

b) graph\_colouring()

Return: Writes to a CSV file and returns it.

graph\_modeling()

Input: Takes the data structure from the get\_input() function.

Return: Returns a graph representing various data obtained.

graph\_colouring()

Input: Takes the graph returned from graph\_modelling() function

Return: returns the minimized colored graph.

### 3) Update Timetable

get\_timetable()

Input: Takes CSV file of the generated time table. This is from the return value of generate\_timetable()

Return: CSV file

update\_timetable()

Input: Old timetable CSV file, New updates in another CSV file

Return: New CSV file after updating the old timetable CSV file.

### 4) Authentication of HOD

authentication\_of\_HOD()

Uses 2 helper functions

a) login() and b) user\_validation.

login()

Input: String Name of HOD, String Password

Redirects to extra features of HOD.

Helper functions: a) user\_validation()

Return: void

user\_validation()

Input: String Name of User, String password

Matches name, password with stored values.

Return: Bool. "True" or "False".

5) Grant permissions

approval\_of\_HOD()

Input: File Time table (CSV file)

Return: Bool. "True" for approval and "False" to deny.

6) Display time table

display\_timetable()

Input: Timetable CSV file

Return: Successful display - Bool {"True" or "False"}

7) Report generator

report\_generator()

Input: takes generated timetable CSV File

Return: File: PDF of the workload assigned to teachers, File:

Final timetable PDF file.

### **3.3 Interface Description**

i) Admin and HOD

The input manager (administrator) communicates directly with the HOD to seek approval of inputs.

ii) Student - Time Table interface

The output of the time table generator is a csv file which contains different subjects in diff time slots.

The students can view each of their time tables in the form of a csv file.

## 4. Detailed Design

### 4.1 RTM

Requirement-ID	Requirement Description	Design Component	Test-Case #
3.1	Generate Timetable	Timetable generator	-
3.2	Authentication and Authorization of HOD	Authentication of HOD And Grant permission	-
3.3	Reporting and Analytics	Report generator	-
3.4	Viewing and Accessing Schedules	Display timetable	-