Software Requirements Specification

for

Time-Table Generator

Version 1.0 approved

Prepared by CS-08

Aman Yadav(201651007)

DakshKumar Gondaliya(201651014)

Kirtika Singhal(201651024)

Mayank Pathela(201651029)

Nikhil Sachan(201651034)

Parmeshwar Kumawat(201651035)

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This document provides a detailed description of the requirements for generating the school timetable. It will illustrate the purpose and complete declaration for the development of the system. The timetabling problem involves scheduling a number of tuples, each consisting of a class of students, a teacher, a subject, to a fixed number of time slots. A number of such tuples may be scheduled in the same time slot providing no class or a teacher appears more than once in the time slot.

1.2 Document Conventions

This document has been made in accordance with the standard IEEE format.

1.3 Intended Audience and Reading Suggestions

This document is intended for Developers, Project Managers, Unit testers, Client and future developers.
The rest of the SRS is divided into four sections:

1. The first one provides an overview of the system functionality and introduces different kind of stakeholders and their interaction with the system. It also mentions the system constraints and assumptions about the product.
2. The second section provides a description of various interfaces for the system.
3. The third section deals with various functional requirements.
4. The last section deals with various non-functional requirements. It includes both security and performance related requirements along with the specifications of the attributes which ensure software quality.

It is better to proceed from the description of the product and go step by step for the better understanding of software. If you already have an abstract understanding, then you can skip the introductory part.

1.4 Product Scope

Timetable creation is a time-consuming task. To create a timetable for the school, it takes lots of patience and man-hours. So we are trying to develop a software which is intended to reduce the difficulties and time required for generating a timetable by using constraint programming. The system will take various inputs like details of students, subjects, sections and teachers available, depending upon these inputs it will generate a possible timetable. Once this software gets developed it can be used by any school provided they satisfy all the hard constraints of the designed algorithm.

2. Overall Description

2.1 Product Perspective

The school timetable generator software is currently available in the market but they lag in some or the other features like most of them don't have a User Interface and interactive experience. So the software which is being developed will take care of this and will also present the best possible solution. It is based on the constraints specified by the client for the timetable.

2.2 Product Functions

The major functions are stated below:

- User Interface for filling in all the required information
- Generation of the timetable

2.3 User Classes and Characteristics

The software doesn't require any technical expertise and can be used by the schools to generate a timetable of lectures satisfying all the hard constraints of the algorithm designed. The users are required to fill in all the necessary details for the generation of the timetable and submit it. Then they may have to wait for sometime approx 1-5 mins, depending on the inputs, to get a possible solution.

2.4 Operating Environment

The software will be a web application and can be operated on devices with an internet connection and web browser supporting high RAM space(????). It requires the web browser to be able to support the Web-development framework, i.e, React.

2.5 Design and Implementation Constraints

- The algorithm designed to provide a solution to the timetable problem has to follow the constraints listed below:

a) **Hard Constraints:** Constraints that cannot be violated while a timetable is being computed. These are:

1. A section of a class should have only one lecture at a time slot.
2. The teacher should not have more than one lecture at the same time slot.
3. There should be no free period in between the lectures for a class of students.
4. All the lecture hours should be of the same duration and the lab hours duration longer than each lecture hour and must be fixed in the beginning.
5. Each class has a fixed number of lectures in a day. **(half-days are not considered)**

b) **Soft Constraints**: The constraints that are desired to be addressed in the solution as much as possible. These include:

1. A teacher should have at least one free time slot in a day.
2. Any teacher is allowed at most ‘k’ number of lectures in a week. The value of k is accepted before the execution of the algorithm.
3. Two subsequent classes of one subject should not be scheduled; if necessary, only once a week

- **Time and Memory constraints**: The software may take a long time, say approx 4-5 mins to execute and requires high RAM in order to perform iterations for finding a possible solution.
- Logging will be maintained for the admin.
- We are going to use Amazon cloud service(AWS) for the database.

2.6 User Documentation

There will be an FAQ section in the site which will cover most of the questions the admin will need. There will be a separate User Manual for the same.

2.7 Assumptions and Dependencies

Assumptions are:

1. All sections of a class have the same capacity.
2. Teachers are available for the whole day and week, i.e, no preferences are taken into consideration.
3. The number of subjects needs to be finalized before the algorithm begins execution.
4. The number of teachers entered before the execution of the algorithm is assumed to be constant.
5. Classrooms for any section fixed throughout the day.
6. Every day in the week is assumed to have the equal number of time slots.

3. External Interface Requirements

3.1 User Interfaces

- The application should be user-friendly. It should not be overloaded with unnecessary icons and information.
- The interface should guide the user to the appropriate screens so that user does not feel lost in the middle of the application.
- The information once filled by the user will be stored into the database and the user can retrieve from there for next time generation. Also, the user has the choice to modify some entries or clear the database through the click of a button.

- If an event occurs, a dialog box will be displayed on the screen informing the user whether the event is successful or there is some error.
- A login page for the admin to keep the database values to be modifiable by admin only.

3.2 Hardware Interfaces

The software which is a web application can work on devices with high RAM space and fast processors provided there is an internet connection and a browser.

3.3 Software Interfaces

Our product is a web application so it would be mainly using three other software components:

- Server
- Database
- React Framework

The server is required for deployment of the web application to be available online for everyone.
The database queries will be called at respective times when the user wants to modify the data or retrieve some information.
The web framework will help to design the User Interface.

3.4 Communications Interfaces

The communication functions required by the software are:

- a well-supported web browser
- a network server using HTTP protocol.

4. System Features

This section includes the requirements that specify all the fundamental actions of the software system.

4.1 System Feature

4.1.1 Description and Priority

The main aim of our application is to provide the best possible solution to the timetable problem. For this, the User Interface is designed keeping in mind the requirements of the admin and other teachers.

4.1.2 Stimulus/Response Sequences

Admin's Side

1. Admin needs to register or log in and then login into the application.

2. Now the admin has two options: to create a timetable by filling in all the details and to view the timetable and edit it.

Teacher's side

1. All the teachers will be provided with the same password and they need to log in through that.
2. After logging in the teacher has the only option of viewing the timetable.

4.1.3 Functional Requirements

The functional requirements are:

- ID: TT0

  Title: Home Page

  Our application has provision for two types of users: Admin and Teachers. This is a page which leads the user to respective interfaces and for this, there is an option given to select which type of user is the person accessing the website.

  User Class 1: Admin

- ID: AD1

  Title: Sign Up

  Description: Before using the application for generation of timetable the admin has to sign up and log in. It is of high importance as the information regarding the teachers, subjects, and others is stored in the database and can be retrieved and modified, only by the admin.

- ID: AD2

  Title: Login

  Description: The admin needs to log in into the application. A dialog box will be displayed informing whether successfully logged in or some entries don't match.

- ID: AD3

  Title: Admin Selection Page

  Description: The admin needs to select whether he wants to generate the timetable, view or modify the timetable entries.

- ID: AD4
  Title: Details
  Admin needs to enter the details of subjects, teachers, electives, and labs on this page.

- ID: AD5
  Title: 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 dropdown list will reduce the efforts of typing the name of a teacher again and again.

- ID: AD6
  Title: View/ edit timetable
  The admin can view the generated timetable and can modify it.

User Class 2: Teacher

- ID: T1
  Title: Login
  Description: The teacher needs to enter the username and password which will be same for all the teachers.
- ID: T2
  Title: View timetable
  Description: The teacher can view the complete timetable as well as the slots at which he has lectures.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- In a situation where there is no internet connection, an empty home view should be prepared to display the inability of the app to fetch the required data.
- The application should be such that it is easy to understand its working and to operate the application. The UI design has been made such that it is attractive and user-friendly.

5.2 Safety and Security Requirements

Software application should provide a secure login, registration for the teachers and admin. It should make sure that these details can be changed. Session management should be established and should be ended while logging out (after the task is done).

5.3 Software Quality Attributes

- The system should be cost-effective to maintain things. Maintainability requirements may cover various levels of documentation such as system documentation and documentation related to testing and test cases.
- The reliability of the application depends on the constraints and the data provided.