Software Requirements Specification

for

TimeTable Generator

Version 2.0 approved

CS -08

Indian Institute of Information Technology Vadodara

Team Members

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
Kirtika Singhal	16/09/2018	Initiate	1.0
Mayank Pathela	20/09/2018	Add points	1.1
Aman Yadav	24/09/2018	Changes made after pivoting to School Time Table	2.0

1. Introduction

1.1 Purpose

This document provides a detailed description of our software application, which is a School Timetable Generator. We have proposed a heuristic solution. Hence, our solution may not be optimal, nevertheless, it will be close to the optimal solution. The software solution is generic. This severely increments the scope of our application as it will be an open source software and could be used by any individual or institution. This document will illustrate the functional and non-functional requirements of the software solution. It will define the system features, product description and the dependencies of the software.

1.2 Document Conventions

This document has been made as per the IEEE standards[1].

1.3 Intended Audience and Reading Suggestions

This document is intended for Developers, Project Managers, System Testers, Client and future collaborators.
The rest of the SRS is divided into four sections:

1. The first one provides an overview of the system functionality and introduces a 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 describes various external interfaces for the system. It includes the user, hardware, software and communication interfaces.
3. The third section deals with various functional requirements of the software. It includes a detailed description of the system features.
4. The last section describes various non-functional requirements. It includes the security and performance related requirements along with the specifications of the software quality attributes.

1.4 Product Scope

Timetable creation is a time-consuming task. While creating the timetable for a school, it takes lots of patience and man-hours of the management and faculty. This massive amount of human efforts consumed by time-table generation can be invested in more productive activities. Hence automation of this process is the solution. Accordingly, we will develop a software solution which will reduce the complexities and time consumed in generating a timetable on the user side.

[1] The IEEE Software Requirements Specification Document - http://home.agh.edu.pl/~jsw/io/IEEE830.pdf

This application will also work as an online portal for the school employees to view the timetable. Moreover, it will also keep data infeasibility in check.

2. Overall Description

2.1 Product Perspective

The school timetable generator software is currently available in the market, but the software is high priced and there is no guarantee if it would function desirably. Such a software can be ineffective if it doesn't consider the exact constraints of the user. The developed software is based on the requirements specified by the client. It will present a valid solution or atleast a solution close to the valid one.

2.2 Product Functions

The primary functions are stated below:

- Generation of the timetable.
- User Interface for filling in all the required information
- An interface to view the data before timetable generation. The admin can also change the invalid data if any.
- An interface for the teachers and admin to view the generated timetable.
- The teachers and Admin can also view the individual timetable of a teacher.

2.3 User Classes and Characteristics

The software doesn’t require any technical expertise and can be used by the school management to generate a timetable satisfying all the hard constraints of the algorithm designed. It requires basic knowledge of computers i.e. the Admin needs to know how to login into a website and input data in the form fields. The users are required to fill in all the necessary details for the generation of the timetable and submit it to get the possible timetable 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. It requires the web browser to be able to support the Web-development framework React.js. The popular browsers which will support it are - Chrome, Firefox, Safari, edge etc.

2.5 Design and Implementation Constraints

- We are using constraint-based programming. The software solution will take an input of the time slots and will randomly try to fill the slots. If a valid timetable is generated, the

software will present that as the solution. Otherwise, it will regenerate until a valid solution for the timetable is found. If no valid solution is found, then after a predefined number of regenerations, the software will display the best fit found.

- The algorithm designed to provide a solution to the timetable problem based on the input data which has to follow the constraints listed below:

a) **Hard Constraints:** Constraints that cannot be violated by the input and output data. All the hard constraints concerning the input data have to be fulfilled by the input data. 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 between the lectures for a class of students.
4. Each class has a fixed number of lectures in a day.
5. The number of lectures a teacher will have in a week should not be more than the total number of time slots.
6. The number of lectures a section will have in a week should not be more than the total number of time slots.

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.
3. The free slots of a faculty should be spread across all days. For e.g. if a faculty has 6 free slots in a six day week, then there should be one free slot every day.

- **Time and Memory constraints**: The software may take a long time, say approx 3-4 mins to execute and requires high RAM space in order to perform the iterations required for finding a possible solution.
- Logging will be maintained for the user.
- We are going to use Heroku[2] for website deployment.

2.6 User Documentation

There will be a separate Handbook (User Manual) for the website. It will specify how a user can interact with the user interface and generate the timetable.

2.7 Assumptions and Dependencies

Assumptions are:

1. Except for electives, classrooms for any section is fixed throughout the day.
2. It is assumed that all sections have a specific classroom and these classrooms are of the required capacity to accommodate the section. Hence the classroom and section are considered the same entity.

[2] Heroku is a cloud platform used to deploy website. Here is the link.

3. Teachers are available for the whole day on all working days of the week, i.e, no explicit preferences are taken into consideration.
4. The number of teachers entered before the execution of the algorithm is assumed to be constant.

3. External Interface Requirements

3.1 User Interfaces

- The application will be user-friendly. It will not be overloaded with unnecessary icons and information.
- The interface will guide the user to the appropriate screens so that the 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 can also be used for timetable generation in future logins. 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. An event can creating a user profile or be generating the timetable.
- There will be a login page for the admin, to ensure that only the admin can modify the database and generate timetable.

3.2 Hardware Interfaces

The software will be a web application and therefore it will work on devices with an internet connection and a browser. The ram requirements are greater than or equal to 4 GB.

3.3 Software Interfaces

- Our product is a web application so it would mainly be using three software components:
  - Server
  - Database
  - React Framework

- The database will be created using MongoDB. It will store the user data, input data and the generated timetable.
- The ReactJS framework will be used to design the User Interface.
- The server will be created using the Express.js Framework. It is used to link the frontend (React Application) to the backend (Mongo Database).

3.4 Communications Interfaces

The communication interfaces required proper functionality by the software are:

- a react supported web browser. The browser should also support the IP and HTTP protocols
- An internet connection of at least 500 Kbps download speed.

4. System Features

This section includes the features that the software system will include. It describes the system functionality.

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

1. Admin needs to register or log 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.
3. The teachers can only view the timetables.

4.1.3 Functional Requirements

The functional requirements are:

- ID: TT0

  Title: Home Page

  The application will have 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: User

- ID: AD1

  Title: Sign Up

  Description: Before using the application for generation of timetable the admin has to sign up. It is of high importance as the information regarding the teachers, subjects, and sections are 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 dialogue box will be displayed informing whether successfully logged in or some entries (password,username) 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: Enter Details
  Admin needs to enter the details of subjects, teachers, electives, and labs on this page.

- ID: AD5
  Title: Create Mappings

  This page will display the entered data. It will also notify the admin if there any data point is breaking any hard constraints.
  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 or section repeatedly.
  This page will also contain the "generate" button, which will be used by the admin to generate the timetable.

- ID: AD6
  Title: View/ edit timetable
  Through this page, 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 their timetable and the timetable of other teachers. They also have the option to view the timetable section wise.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- In a situation where there is no internet connection, a notification should be prepared to display the inability of the app to fetch the required data.
- The application will be such that it is easy to understand its working and therer is ease of operation. The UI design will be attractive and user-friendly.

5.2 Safety and Security Requirements

The software application will provide a secure login, registration for the admin. It will make sure that the database can be changed only by the admin. Session management should be established. The session will be when the user logs out or after 6 hours of inactivity.

5.3 Software Quality Attributes

- The system software will be a cloud deployed web application, hence no maintenance issue on the user side.
- All the functionality specified will function as they have been described.
- There will be no security issue.
- The system will notify if a hard constraint is broken by the input data and will not display any output data violating the hard constraints.
- As all the data is stored on the database, data (input or output data) reusability is a major plus point.