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

1.Introduction **4**
1.1.Purpose 4
1.2.Document Conventions 4
1.3.Intended Audience and Reading Suggestions 4
1.4.Product Scope 4
2.Overall Description **5**
2.1.Product Perspective 5
2.2.Product Functions 5
2.3.User Classes and Characteristics 5
2.4.Operating Environment 5
2.5.Design and Implementation Constraints 5
2.6.User Documentation 6
2.7.Assumptions and Dependencies 6
3.External Interface Requirements **7**
3.1.User Interfaces 7
3.2.Hardware Interfaces 7
3.3.Software Interfaces 7
3.4.Communications Interfaces 7
4.System Features **8**
4.1.System Feature 8
4.1.1 Description and Priority 8
4.1.2 Stimulus/Response Sequences 8
4.1.3 Functional Requirements 8
5.Other Nonfunctional Requirements **9**
5.1.Performance Requirements 9
5.2.Safety and Security Requirements 9
5.3.Software Quality Attributes 10

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

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

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

which will reduce the complexities and time consumed in generating a timetable on the user side. 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 a 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 at least a solution close to the valid one. It is based on the constraints specified by our 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 satisfying all the hard constraints of the algorithm designed. It requires no knowledge of computers and is designed to allow you to sit at the keyboard controls and 'drive' your way through the timetable. The users are required to fill in all the necessary details for the generation of the timetable and submit it in order 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, i.e, React.

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 time period, the software will display the best fit found.

- 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 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 or equal to each lecture hour and must be fixed in the beginning.
5. Each class has a fixed number of lectures in a day.

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 3-4 mins to execute and requires high RAM space in order to perform iterations for finding a possible solution.
- Logging will be maintained for the user.
- 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 is fixed throughout the day.
6. Every day in the week is assumed to have an 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 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 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 dialogue 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 will be a web application and therefore it will work on devices with an internet connection and a browser.

3.3 Software Interfaces

Our product is a web application so it would mainly be using three 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

1. The user needs to register or log in and then login into the application.
2. Now the user has two options: to create a timetable by filling in all the details and to view the timetable and edit it.

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: User

- 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 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 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 it. The UI design will be attractive and user-friendly.

5.2 Safety and Security Requirements

Software application should provide a secure login, registration for the teachers and admin. It will make sure that the database can be changed only by the registered users.

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.