

---

# High-Level Design

for

## Time-Table Generator

Version 1.0

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

<b>Introduction</b>	<b>3</b>
<b>1.1 Purpose</b>	<b>3</b>
<b>1.2 Scope</b>	<b>3</b>
<b>1.3 Overview</b>	<b>3</b>
<b>General Description</b>	<b>3</b>
<b>2.1 Product Perspective</b>	<b>3</b>
<b>2.2 Tools Used</b>	<b>4</b>
<b>2.3 General Constraints</b>	<b>4</b>
<b>2.4 Assumptions</b>	<b>4</b>
<b>Design Details</b>	<b>4</b>
<b>3.1 User Interface</b>	<b>4</b>
<b>3.2 Use Case Diagram</b>	<b>5</b>
<b>3.3 Software Interface</b>	<b>5</b>
<b>3.4 Hardware Interfaces</b>	<b>6</b>
<b>3.5 Performance Requirements</b>	<b>6</b>
<b>3.6 Safety and Security Requirements</b>	<b>6</b>

## Revision History

Name	Date	Reason For Changes	Version
DakshKumar	30/09/2018	Initial Document	1.0

# 1. Introduction

## 1.1 Purpose

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding and can be used as a reference manual for how the modules interact at a high level.

## 1.2 Scope

The HLD documentation presents the structure of the system, such as the application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

## 1.3 Overview

The HLD will:

- present all of the design aspects and define them in detail
- describe the user interface which will be implemented
- describe the hardware and software interfaces
- include design features and the architecture of the project
- list and describe the non-functional attributes like:
  - ❑ Performance requirement
  - ❑ Safety and Security Requirement

# 2. General Description

## 2.1 Product Perspective

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. So, TimeTable Generator software will reduce the complexities and time consumed in generating a

timetable on the user side. 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.

## 2.2 Tools Used

1. Nodejs will be used for creating our back-end side server
2. MongoDB to store our data.
3. mlab AWS-Sandbox cloud storage to store all of our user data.
4. React for front-end development
5. JWT-authentication to manage different user and their data. Once the user is logged in, each subsequent request will include the JWT(JSON Web Token), allowing the user to access routes, services, and resources that are permitted with that token.

## 2.3 General Constraints

The TimeTable Generator must be user-friendly and as automated as possible. Administrators should not be required to do anything besides the initial setup, and users should not be required to know any of the workings. Without logging in, the user will only have the ability to view the home page and the links to pages in the navigation bar. After logging in, that user then has the ability to access the database.

## 2.4 Assumptions

This project is based on the idea of automating the process of generating the school timetable, and the goal is to make this idea a reality using Software Engineering practices. In doing so, many documents are created, and it is assumed that design flaws will be found early on. It is also assumed that all aspects of this project have the ability to work together in the way the designer is expecting. The second assumption is that a demonstration and presentation will be possible in the mid-November.

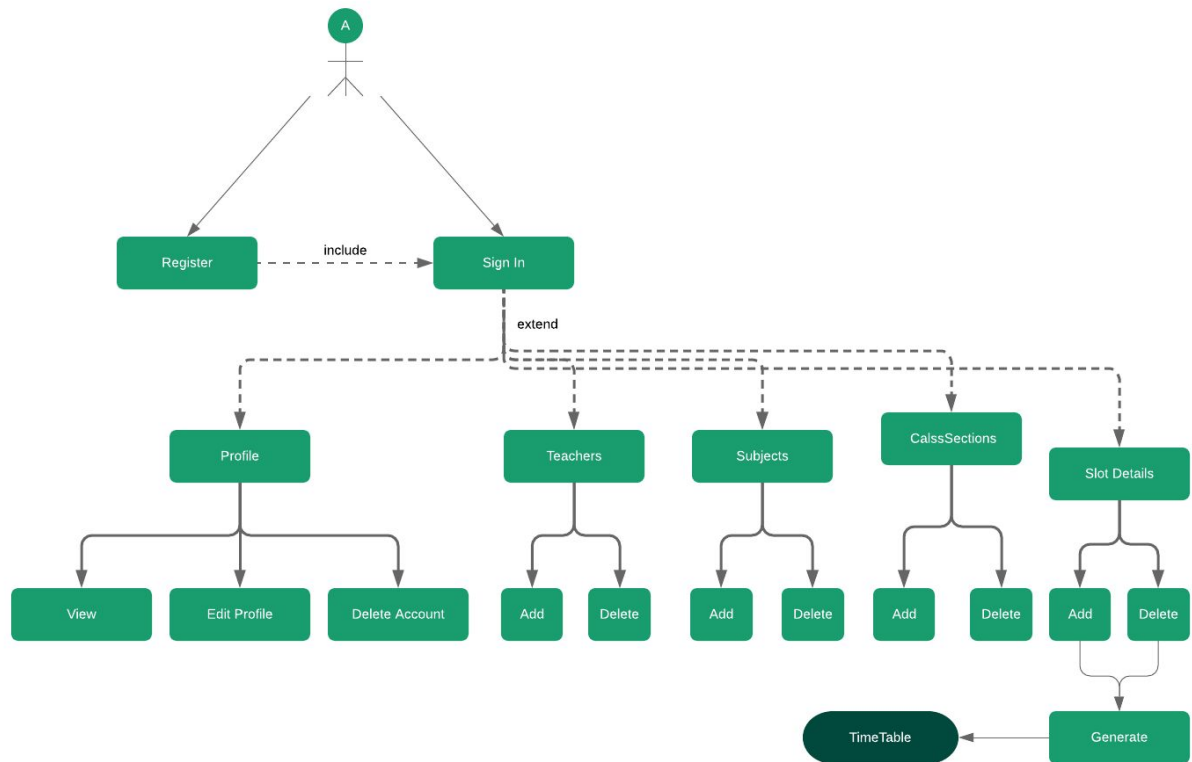
# 3. Design Details

## 3.1 User Interface

The user interface is a very simple plain layout with little to no graphics. It will display information very clearly for the user and will primarily output information to the user through HTML pages. Screens are used mainly for input through text fields in HTML

pages. Screenshots have been provided to demonstrate the user interface in a separate document 'User Interface Prototype'.

### 3.2 Use Case Diagram



### 3.3 Software Interface

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 Hardware Interfaces

The software will be a web application and therefore it will work on devices with an internet connection and a browser.

### 3.5 Performance Requirements

- The system must be interactive and the delays involved must be less. So in every action-response of the system, there are no immediate delays.
- 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.

### 3.6 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).