Software Requirements Specification

for

DigiSchool

Version 1.1

Prepared by

Group #: 6 **Group Name: Dementors**

GUGULOTH VARUN	180272	varunguguloth@gmail.com
SUMIT NAGLE	170732	sumit.nagle@gmail.com
GAURAV KUMAR	180265	phogat102135@gmail.com
MOTA JITENDRA	180434	jitendratherider23@gmail.com
AKASH DAYANAND CHAVAN	180054	akashchavan180054@gmail.com
DIKSHANT RAJ MEENA	180247	dkrajmeena27899@gmail.com
SHREYASI PRASAD	190824	student.shreyasi@gmail.com
YASHWANT TAILOR	180894	yasht88101@gmail.com
YEDDULA VINAY CHANDU	191181	chandu8042439@gmail.com
AMAN RAW	190108	amanraw4444@gmail.com

Course: **CS253**

Mentor TA: **Shatroopa Saxena**

Date: **30-01-2022**

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
V1.0	Sumit, Varun G	first draft	
V2.0	Shreyasi Prasad	All the performance characteristics and safety features are added	

1 Introduction

1.1 Product Scope

DigiSchool is a digital platform which focuses on the academic sector. Objective of this product is to provide a platform where a student/teacher gets the same environment as they get in the school/college, however digitally.
DigiSchool mainly focuses on schools, but also keeps a positive goal of providing the same robust facilities of our application to universities and colleges.

Learning digitally, gives students access to course material throughout the duration, which helps students to study better and they don't have to worry about losing any course content and can always be in phase with all other students.
Moreover, situations like pandemic don't halt their learning, as all the course material is available to students 24X7 through smart phone or laptop.

1.2 Intended Audience and Document Overview

Documents contain the information about our web applications, and the content of this document targets users, stakeholders, project managers and developers.

Users, project managers and stakeholders should start reading this document from section 2.1, and the features they mostly ponder are in section 3.
Developers should read the whole document thoroughly as it contains all the system and user requirements as well as functional and nonfunctional requirements that our web application should have.

1.3 Definitions, Acronyms and Abbreviations

IEEE : Institute of Electrical and Electronics Engineers

1.4 Document Conventions

This document follows IEEE formatting requirements with slight modification, in which we are using Italic Arial font of size 11 throughout.
Whereas important terms are marked with red ink. Moreover, we use the underline, in order to phrase out important statements that the reader should focus on.
Any comments are inked with blue.

1.5 References and Acknowledgments

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. >

2 Overall Description

2.1 Product Overview

DigiSchool comes in the category of EdTech softwares. We took an approach of developing web application from scratch. We choose web applications over desktop applications, because any device (laptop or phone) running on any modern OS (linux or windows) which has the capability of using a modern browser can easily use our application from anywhere in the world.

DigiSchool is a self-contained product which can be accessed by any user who has a phone or computer with a reliable internet connection.
It is a platform with multi-user interface, and provides a one-stop platform for all the academic-related activities such as class monitoring, quiz and assignments, student engagements and performance analysis, student interactions via forums.
It is a highly interactive interface and provides easy-to-navigate interface for users.

2.2 Product Functionality

We believe that learning should be safe and secure while over pondering with all the features that the students and teacher seek to have available all together on a single platform. And so we hear what they want and DigiSchool provides the functionalities such as,

- *Login for authentication with OTP verification, which ensures the authenticity of the user. And with our backend mechanism of recognition, we identify the specific user, and then, we populate our web pages accordingly. As of now, we have only three privileges levels, teacher, student and admin.*
- *Discussion, where users can interact with all the other users (including teachers) enrolled in the same course, and make the learning even more better.*
- *Test, provides all the necessary assessment techniques (quiz and assignments), so that students and teachers can assess the extent of their learning.*
- *Lectures, contains the recorded video lectures.*
- *Announcement, and alert page where teachers keep all the students up-to-date about new news and events.*

2.3 Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and ed; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software). You can be creative here to some degree.>
We implement our application as a web application, thus any user can access our application, via any modern browser, thus, hardware limitation comes from the browsers perspective, and for that you can refer to it here. In summary, the hardware limitations are,

- *Processor: 1. GHz +*

- *minimum RAM: 128 MB*
- *OS: Windows XP, Vista, 7+, Mac OS X 10.5+, or Ubuntu 10+*
- *Internet connection with 1+ Mbps (although lesser speed is ok, but the website loading experience will not be that good)*
- *Browsers must support ssl or tls.*
- *Browser must support all the HTTP request and response method. As it should implement file sending.*
- *Browser must send all the meta and header data such as cookies and tokens.*

2.4 Assumptions and Dependencies

Assumptions:

- *The user's browser supports the minimum required version of all the utilities used by our application such as language versions.*
- *The browser is using intended activities and nothing malicious, which may tempered the data our application provides to the browser.*
- *There is proper communication setup between the web hosting service and our backend application, ensuring a smooth workflow.*
- *User's web browser supports HTML5, CSS and JavaScript.*

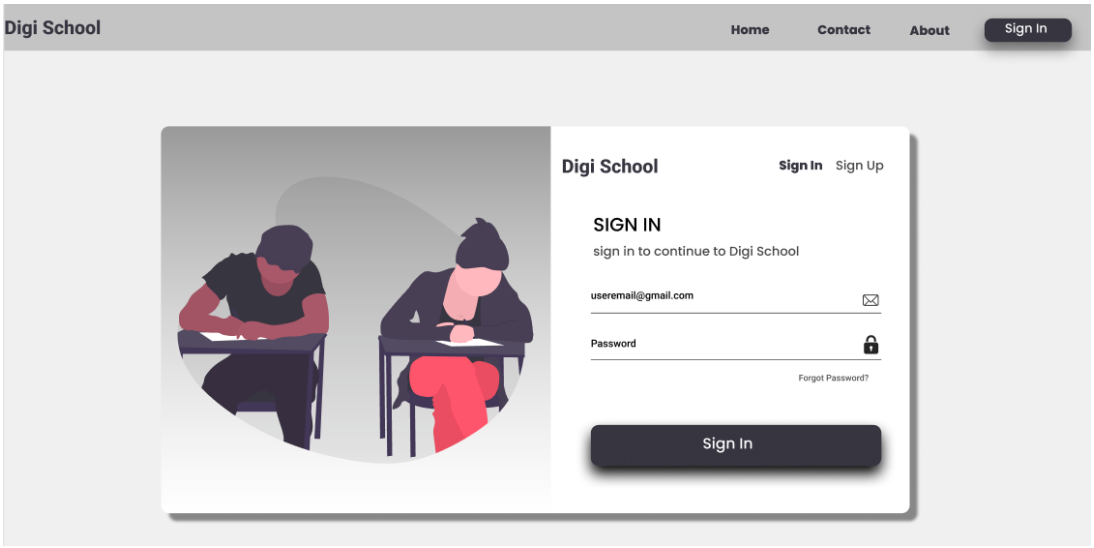
3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

There should be two types of users, Teacher and Student.
All the users should be able to login using correct credentials or create a new account if they are a new user. In creating a new account the user should be able to select atleast one subject for their profile.
All the users should be able to edit their profiles only once in the lifetime of the profile. All the users should be able to view their personal profiles on login.
All the users should be able to Create a discussion topics on the forum on which every user should be able to view and comment.
Teacher user should be able to create and edit lectures for a subject by adding lecture video and documents related to it. Every member in that subject should be able to view that lecture.
Teacher type users should be able to create and edit tests in which they should be able to add instructions and as many questions as they need. These tests should be available to be attempted by Student users in that subject after the start time and till the end time.
Student users should not be able to open the test before start time and after end time. Teacher users should be able to grade the test attempts of the students.
Teacher type users should be able to create and edit announcements for a subject which should be viewable to all the users in that subject.
Some snippets of the expected interface are given below.

Sign In Page:

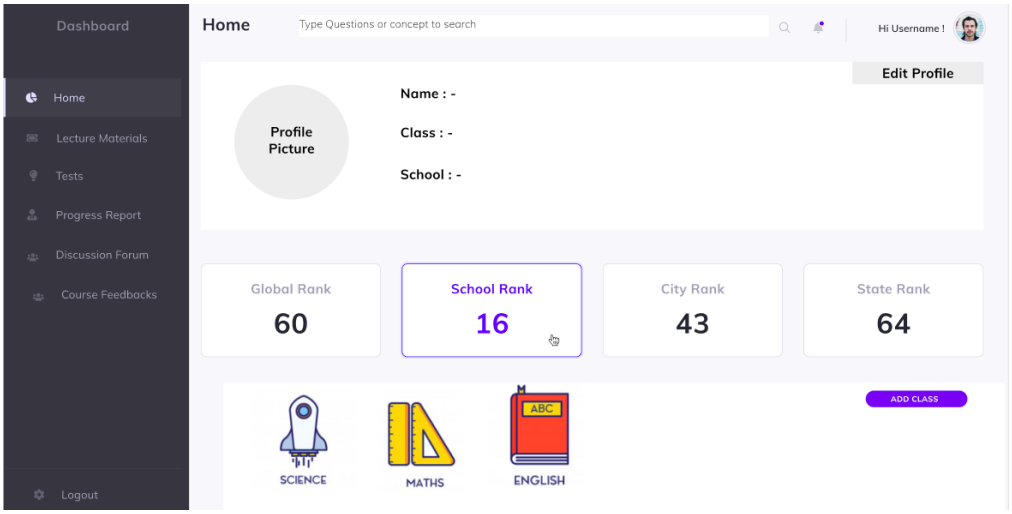


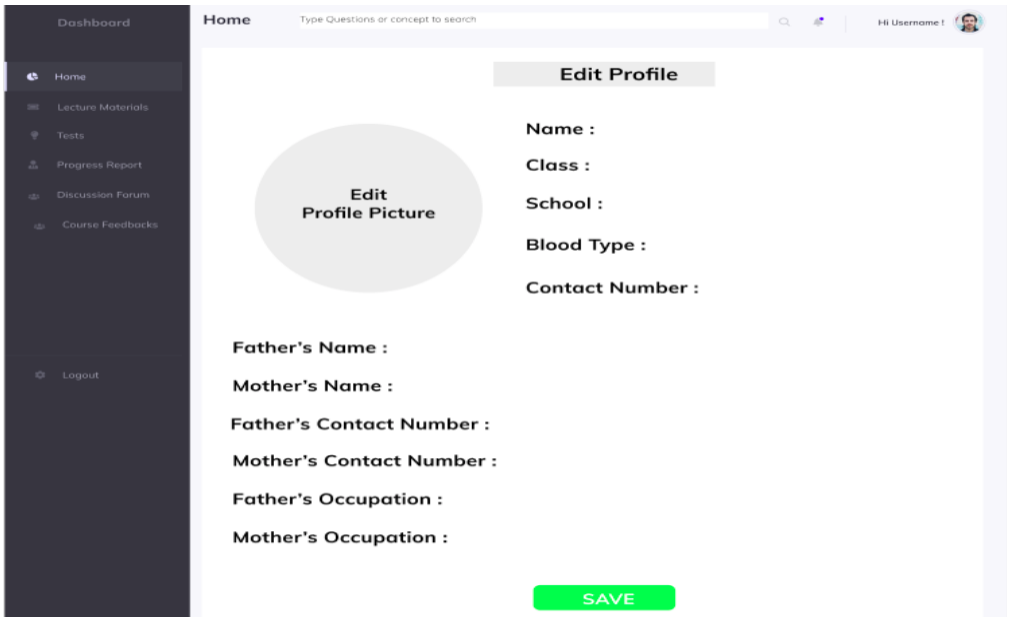
Here users need to provide their login credential, and based on the privilege level of the user, the signed in pages will be reflected.

Sign Up Pages:

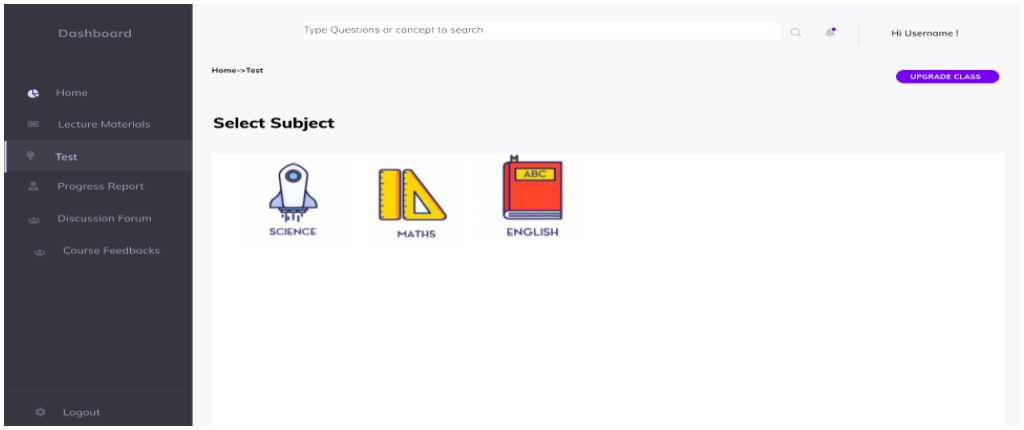
After that based on class and section, the dedicated course list will be visible on the screen.

Dashboard:





Test:



3.1.2 Hardware Interfaces

Since this is a web based application, any device that is connected to the internet and has a web browser that supports HTML, CSS and Javascript (preferably laptops and PCs) shall be used to access this.

3.1.3 Software Interfaces

Web browsers like chrome, firefox , Edge or any market leading web browsers which supports javascript, session cookies shall be used to access the web page.

3.2 Functional Requirements

3.2.1 Authentication for Login and SignUp:

On Login the application should verify username and password and then direct to profile page if it passes else retry login and give a message that username or password is incorrect.
On SignUp the Email should be verified by the use of OTP verification.
The password input on signup shall not be less than 8 characters.
Users shall provide First name, Last name, email, Registration ID, mobile number, school, class and section at the time of signup.

3.2.2 Create, View and Comment on Forum:

There should be a forum where users can communicate by posting doubts regarding a subject and commenting on these posts.
A post should be visible to every user in a subject in which it is posted.

3.2.3 Create, Edit and View Lectures:

A teacher should be able to upload a video and/or files for a lecture and provide its title and description in a subject.
Every user in that subject should be able to view that lecture's content.

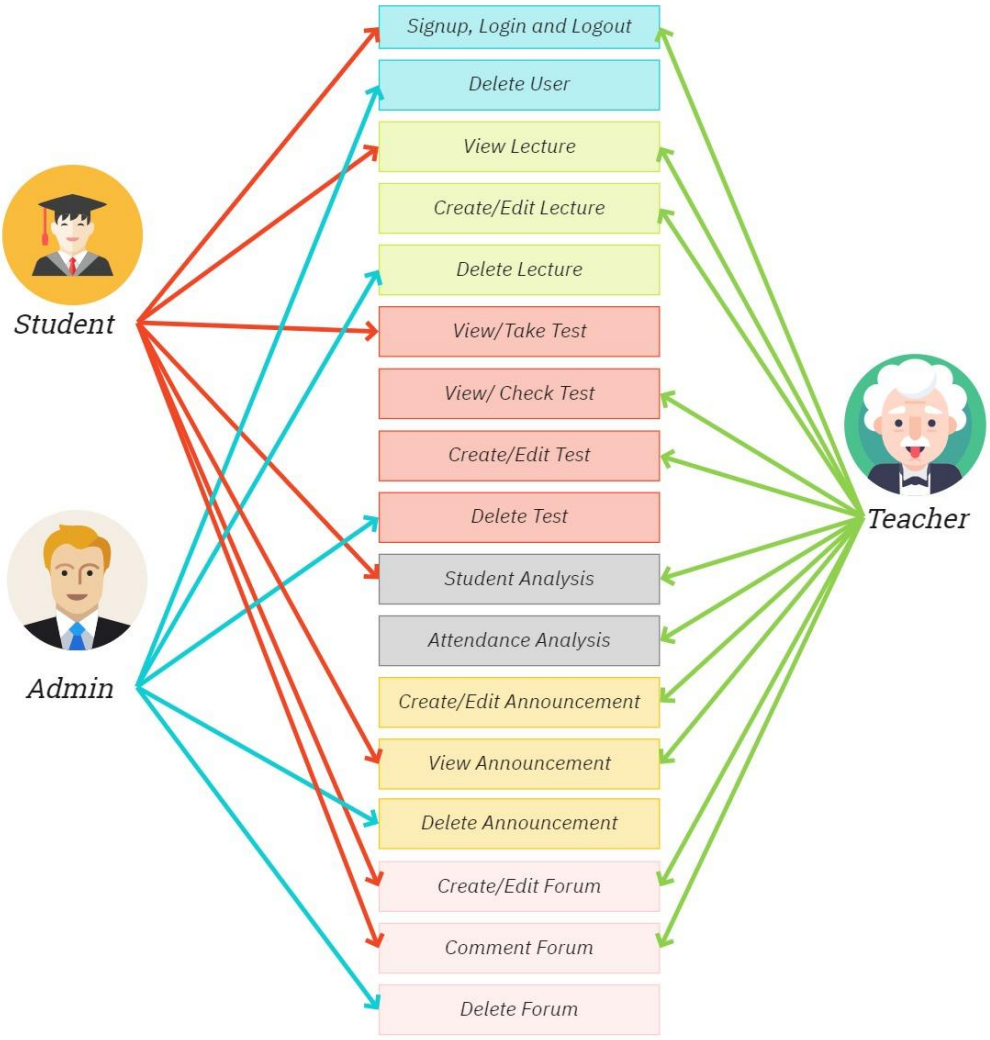
3.2.4 Create, Edit, Attempt and Grade Tests:

A teacher should be able to create and edit a test. While creating a test a teacher should provide a set of instructions for the students to read and follow while attempting the test. There should be no restriction to the number of questions a teacher can add in a test. The teacher should add a start time and an end time for the test. The start time should be atleast one day after creating the test.
Every student in the subject should be able to attempt the test and answer each question in the test in between the start and end time. No user should be able to attempt the test before start time or after end time.
Teachers should be able to see the attempt made by every student and they should also be able to grade the answers.

3.2.5 Create, Edit and View Announcements:

A teacher should be able to create and edit an announcement in a subject. While creating the announcement the teacher should provide a title and description for it. Every user in that subject should be able to view that announcement.

3.3 Use Case Model



3.3.1 Use Case #1 (use case name and a unique identifier – e.g. U1)

Author – *Sumit Nagle*

Purpose *- This use case talks about the use case for teachers and students who will be using the web application.*

Requirements Traceability – *Teacher should have a valid email address and it should be same as the email address given to us by school. Teacher must be enrolled as a teacher in atleast one course.*

Student should be enrolled in atleast one class.

Priority *- Medium*

Preconditions *- Teacher and student should have a valid email address and must have all data inhand such as school id, class section.*

Post conditions *- The teacher and students should able to do the desired functionality as stated in functional requirements.*

Actors *– Teacher and students both are human actors.*

4 Other Non-functional Requirements

4.1 Performance Requirements

- *With a 6-10 MBps internet speed the average page load time shall be less than 5 seconds.*
- *With a 3G or 4G internet connection the application should be able to stream videos without much buffering so the user can get a smooth streaming experience.*
- *While a user is attempting a test the page should not reload unless requested by the user. And all the data must be saved in the frontend itself. In order to avoid rewriting the data.*

4.2 Safety and Security Requirements

- *The application should provide OTP verification for E-mail id while SignUp. The minimum password length for any user should be 8 characters and passwords should be encrypted by hashing and salting while being saved in the database.*
- *CSRF tokens shall be used to prevent the users against CSRF attacks.*
- *All the inputs given in any type of form should be validated.*
- *Privilege checks should be done at various levels verifying that Student type users shouldn't be able to do the functionality of teacher type users.*
- *CIA Triad shall be used to protect data against unauthorized access and malicious attacks.*

4.3 Software Quality Attributes

4.3.1 Adaptability:

Website should be designed such that it can be viewed comfortably on a device of any screen size without getting an error in the user interface. Bootstrap shall be used for this.

4.3.2 Availability:

The website should be accessible to all the users across the internet. Large scale web hosting server shall be used for this.

4.3.3 Reliability:

The website should not be out of service at any point at least while the term for which the users are registered is not complete. The data of the user should stay safe and intact and should not be deleted in any case until the user requests it.

4.3.4 Usability:

The errors occurred during the runtime of the website should be reduced to zero, but in worst case should be atmost 2 per week.

4.3.5 Maintainability:

Group of developers should be available to maintain the website whenever a problem occurs and to give updates to the website.

5 Other Requirements

*<This section is **Optional**. Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>*

Appendix A – Data Dictionary

<Data dictionary is used to track all the different variables, states and functional requirements that you described in your document. Make sure to include the complete list of all constants, state variables (and their possible states), inputs and outputs in a table. In the table, include the description of these items as well as all related operations and requirements.>

Appendix B - Group Log

We had discussions everyday with the team members through whatsapp group/calls and zoom meetings. The team members were also in direct touch with the assigned TA through official Whatsapp Group.

Meeting Minutes	Agenda
Feb 4, 2022 09:30 PM-10:00 PM	Group chat regarding the design.
Feb 5, 2022 04:00 PM-04:45 PM	Group Meet for discussion about various intricacies of the design document.
Feb 6, 2022 10:30 PM-11:30 PM	Group Meet regarding more features that could be added..
Feb 7, 2022 1:00 PM-03:30 PM	Group Meet for discussion on the context diagram and architecture design specifically.
Feb 8, 2022 1:00 PM-1:30 PM	Group Meet for discussion on the class diagram specifically
Feb 10, 2022 05:40 PM-06:30 PM	Group Meet in which each member gave a short presentation about the work assigned and proposed the related reviews.
Feb 11, 2022 05:00 PM-05:40 PM	Group meet regarding work distribution
Feb 12, 2022 12:00 PM-12:30 PM	Group Meet for the discussion on the architecture design
Feb 14, 2022 09:00 PM-11:30 PM	Group Meet to change few UI images for better understanding of the feature
Feb 15, 2022 01:30 PM-02:30 PM	Group Meet to review the work done and to make final updates for this version
Feb 15, 2022 07:30 PM-08:10 PM	Meet for final reviwing