SOFTWARE REQUIREMENTS SPECIFICATION

Paathshaala Website

Version 1.0

Prepared by : 1. Abhishek Bansal (190001001)
2. Arastu (190001001)
3. Dipin Garg(190001013)
4. Harsh Kushwaha(190001018)
5. Jainil Shah(190001020)
6. Prasheel Kumar Tiwari(190004025)

Submitted to : Dr. Puneet Gupta
Assistant Professor

May 2, 2021

Contents

1 Introduction

1.1 Purpose

It is very much difficult to maintain all the data of a course as a hard copy. Any data can be changed, deleted, added at any time. Such as - the admission of new students to an optional course, creation and submission of assignments and quizzes, grading of the students, etc. So, ”PAATHSHAALA WEBSITE” is the solution. The main concept of ”PAATHSHAALA WEBSITE” is to facilitate easy workflow and exchange of information between the professors and the students.

1.2 Intended Audience and Reading Suggestions

This SRS is for developers, project managers, users and testers. Further the discussion will provide all the internal, external, functional and also non-functional information about ”PAATHSHAALA WEBSITE”.

1.3 Project Scope

”PAATHSHAALA WEBSITE” creates a space for the professors and students for maintaining compulsory and optional courses.

After getting admitted to a course a student can view the course information and material. They can also view and submit assignments and view their grades.

Professors will be able to create courses and admit students into them. They can accept the students’ requests of getting admitted into a particular course. The upload of course material and creation of timed assignments and quizzes can also be done through this platform, along with their grading. Lastly, it provides a space for communication among the students and teachers through the discussion forum.

2 Overall Description

2.1 Product Perspective

The system will be operate within university environment. It is a replacement for existing manual hard copy based course management, enabling digitalization of sharing of resources, and day to day activities involved in participation in a course. In future, the app may merge with the university ecosystem, sharing database and keeping an archive of each course.

2.2 Product Functions

1. Professors shall be able to:
    a) Register themselves
    b) Login themselves
    c) Create Courses
    d) Add students to courses
    e) Edit course information
    f) View student applications to join course
    g) Upload course material
    h) Upload timed assignments
    i) View submitted assignments of students and grade them
    j) Create Quizzes for courses
    k) View Students responses for quizzes
    l) Post announcements in discussion thread of a course
    m) View posts of students in discussion thread
2. Students shall be able to:
    a) View Courses enrolled in
    b) Request Courses to be enrolled in
    c) View Course information
    d) View course material uploaded by the teacher
    e) View assignments

f) Submit Assignments before time

g) View grades for assignments

h) Give quizzes set by profs

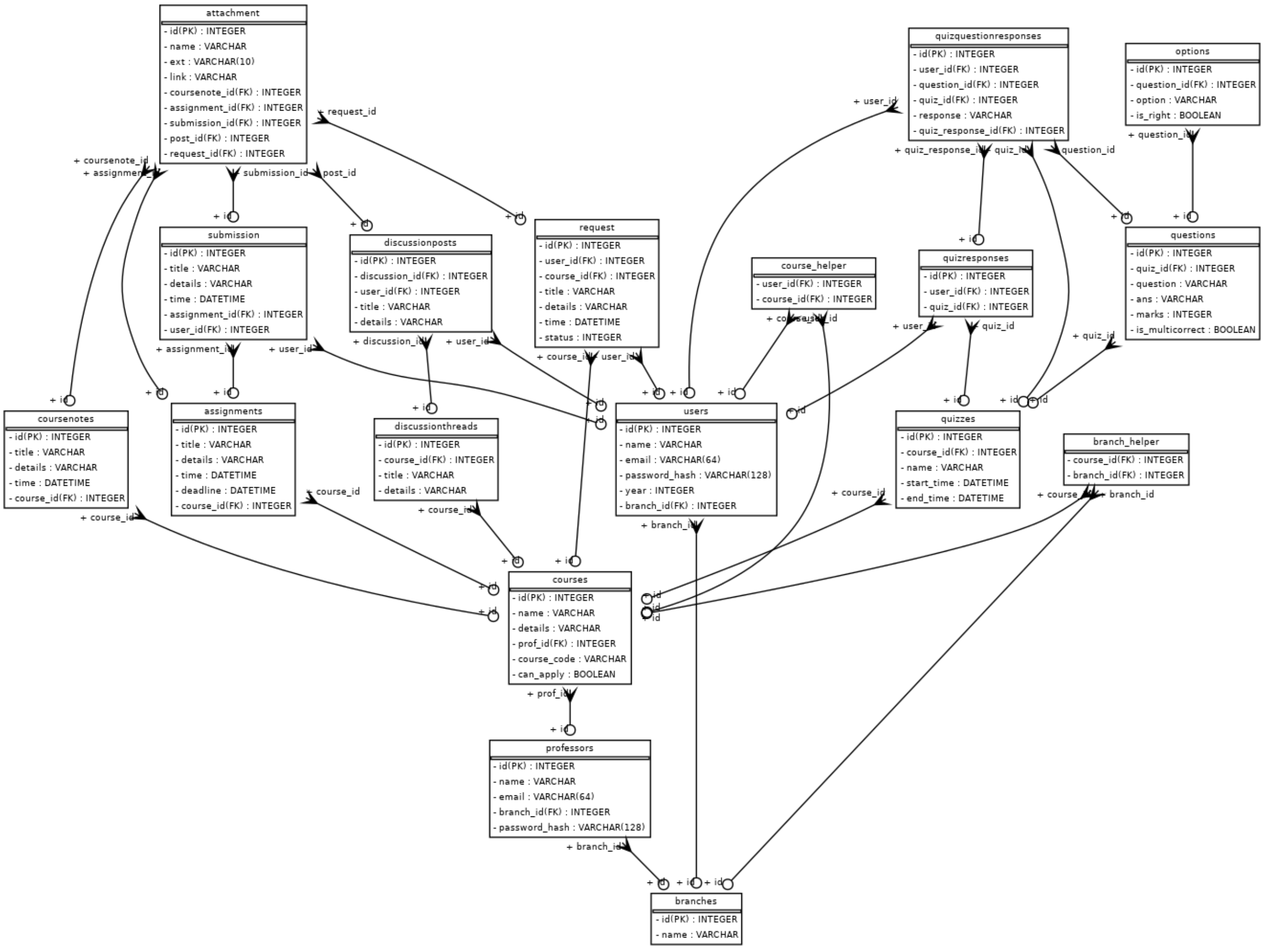
i) View marks for their response in a quiz

j) View posts of students in discussion thread

k) Post in discussion thread of courses they are enrolled in.

2.3 User Classes and Characteristics

View the attached the er diagram above.



2.4 Operating Environment

The website will be operate on all the browsers on any operating environment like windows, mac, ubuntu, etc.

2.5 Design Overview and Website Flow

Professor side website has following flow:

- Professor firsts registers on Professor's Portal
- He then Logs in

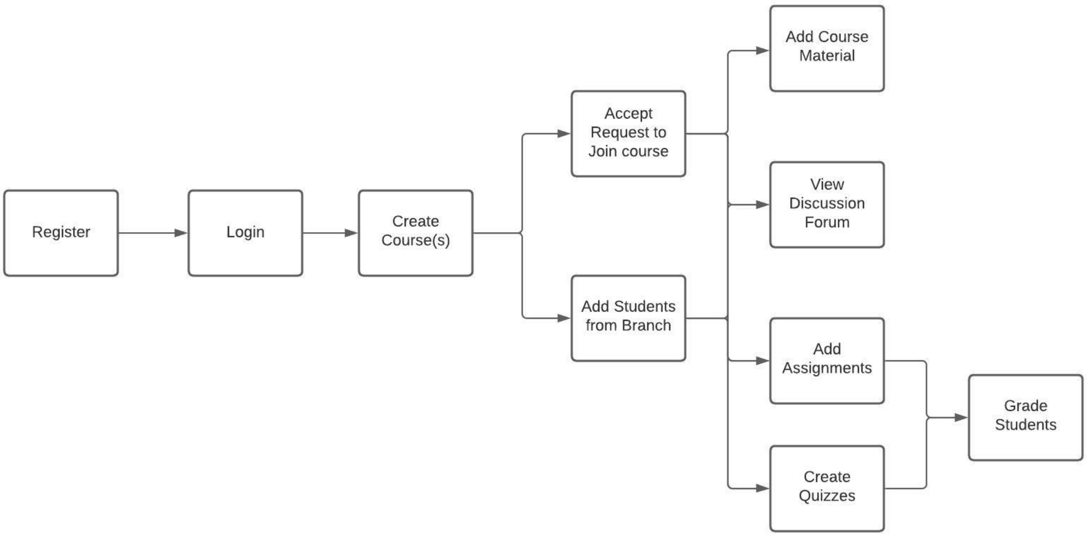


Figure 2.1: Professor Side Website Flow

- Professor can create courses and add students from particular branch or may accept the requests of other students who want to enroll
- He can add study material on the portal and can take part in discussion forum. He can also create assignments and quizzes with fixed deadlines.
- He can grade the assignments and quizzes.

Student side website has following flow:

- Student firsts registers on Student's Portal
- He then Logs in
- Student can enroll in courses available from his department and can request professor to enroll in courses of some other department
- He can download the Study material, posts comments on discussion forum, submit assignments, give quizzes as instructed by Professor or drop a course
- He can view grades on assignments and quizzes.

2.6 Constraints on Design and Implementation

The size of servers bound the application to serve up to 100 clients parallel. This restriction on size of clients may hinder the feature of conducting quiz. Further assumptions are made regarding the compulsion of student to enroll for courses is dictated by the

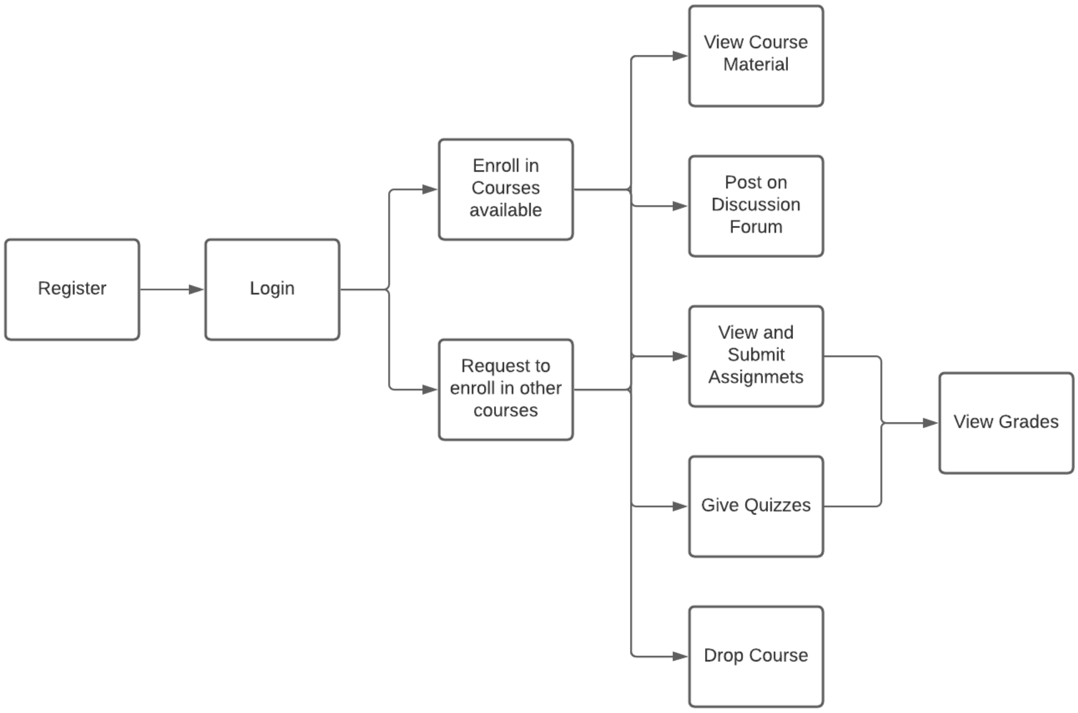


Figure 2.2: Student Side Website Flow

Professor or academic office. The option of non-compulsory courses has been left to students discretion, he/she can drop them if they are not dictated compulsory by the academic office.

2.6.1 Tech Stacks used

The ”PaathShaala” website is being build on Flask framework, Python language, Jinja templating, SQLAlchemy, HTML, CSS and JavaScript.
Back-End - Flask, python
Font-End - Jinja templating, HTML, CSS, JavaScript
Database - SQLAlchemy

2.7 User Documentation

The user documentation will contain a quick start guide, giving details of flow of user controls on website both on side of student and professor. The Admin controls will be also documented and provided to customer.

3 External Interface Requirements

3.1 User Interfaces

The first-time users of the system shall see the log-in page when they use the system. If the users have not registered, they shall be able to do that on the log-in page. Users will have different accessing/viewing/modification privileges based on their account type. The users which are already logged in should be able to see the dashboard page directly when they use the system. Here they can choose which functionality they want to use. The system shall provide a help link on each page to explain how to use the system. The GUI of the product shall be designed in HTML, allowing a multitude of different users access. The HTML design will remove most limits of access because every popular operating system has HTML viewing capabilities.

3.1.1 For students

- The system shall show the students Time Table for the courses which they have enrolled in.
- The system shall allow students to view and fill the course quizzes and assignments.
- The system shall allow students to view course material posted by the course faculty.
- The system shall allow students to discuss with the class problems regarding a course in that course's discussion forum.

3.1.2 For faculty

- The system shall allow faculties to create course and add new students to a course.
- The system shall allow faculties to post course material for a course.
- The system shall allow faculties to create, view, modify quizzes and assignments for a course.
- The system shall allow faculties to discuss on the discussion forums for the courses.

3.2 Hardware Interfaces

All server-side components must execute on server-class computers. All client-side components must execute on personal-class computers. An internet connection and sufficient hardware to run an internet browser are needed to access the files of the website.

3.3 Software Interfaces

The software interface should follow the Model-View-Controller (MVC) model for rendering and modeling data objects. The system shall be web based. The data shall be stored in a database.

3.3.1 OS

- Name: Windows OS
  Mnemonic: Windows
  Version number: 7 or later
  Source: microsoft.com
  Purpose: To allow the system to be used in the user environment.

- Name: Mac OS
  Mnemonic: MacOS
  Version number: 10.4 or later
  Source: apple.com
  Purpose: To allow the system to be used in the user environment.

- Name: Linux OS
  Mnemonic: Linux
  Version number: 14.04 or later(for Ubuntu), 3.1 or later(for Debian)
  Source: linux.org
  Purpose: To allow the system to be used in the user environment.

3.3.2 Web Browser

- Name: Microsoft Internet Explorer
  Mnemonic: IE
  Version number: 11.0 or later
  Source: microsoft.com
  Purpose: To allow remote access of the website and downloading of files via the internet.

- Name: Apple Safari
  Mnemonic: Safari
  Version number: 14.0.3 or later
  Source: apple.com

Purpose: To allow remote access of the website and downloading of files via the internet.

- Name: Opera
  Mnemonic: Opera
  Version number: 10.62 or later
  Source: opera.com
  Purpose: To allow remote access of the website and downloading of files via the internet.

- Name: Google Chrome
  Mnemonic: Chrome
  Version number: 86 or later
  Source: google.com/chrome
  Purpose: To allow remote access of the website and downloading of files via the internet.

3.3.3 Database

- Name: SQLite
  Mnemonic: SQlite
  Version number: 3.29.0 or later
  Source: sqlite.org
  Purpose: To store and access data efficiently and provide facilities to use functionalities on records in the given data.

3.4 Communications Interfaces

The communication among software component shall be performed through message passing over the IP network. From a technical point of view, IP shall be used as the transport protocol. Interactions among surrogates shall be performed using HTTP.

- The system shall send an e-mail message to the student confirming their enrollment in new courses.

- The system shall send a verification email to the new user to confirm the registration process.

4 System Features

4.1 Register and Login

Description: Each Student and Professor are required to register their respective accounts and login to access their authorized data.The interfaces for both these types of users will be different.
Functional Requirements:

- A SQL DB should be used to store login information
- Passwords must not be stored as plain text, rather hashed using bcrypt.
- No two users should be registered with the same email.

4.2 View Courses

Description: On the home page of the student, they can view the courses they are enrolled in and not enrolled in. For professor, they can view the courses they are teaching.
Functional Requirements:

- When viewing courses which the student is not enrolled in, only courses with the same year as that of the student can be viewed.
- Courses enrolled in by the student should be visible in a table along with course info.

4.3 View Timetable

Description: Students (and professors) according to the courses which they are currently enrolled in (or teaching) should be able to view a time-table so that they can manage their time efficiently.
Functional Requirements: The time-table should contain the start and end time of the classes they are currently taking.

4.4 Add and Submit timed Assignment

Description:

- Professors can add timed assignments to their respective courses.

- Students can view the assignments by Professors in courses they are enrolled in.

Functional Requirements:

- Professor should be able to attach files, add written text, set deadline time and date to the assignment.
- Students should be able to access the files and written text of the assignment, and subsequently submit their response before the set deadline.
- The submission is also in the form of attached files or written text.
- The Professor should be able to view responses of all the students.

4.5 Add Courses

Description: Professors should be able to add their courses with specific year and branch.
Functional Requirements:

- Professors can set the name of the course, the year and branch(es) of the course.
- The students of that particular year and branch will automatically be added to the course.

4.6 Request Course Enrollment

Description:

- Students can request the Professor of a course in which he/she is not enrolled in.
- Professor can accept or reject the student based on the application.

Functional Requirements:

- Student can attach files, write a reason in his/her application as to why he want to join this course.
- Professor can view the application files, text to judge the enrollment request.

4.7 Add and View Course Material

Description:

- Professors can add material to their respective courses.
- Students can view the material added by Professors in courses they are enrolled in.

Functional Requirements:

- Professor should be able to attach files, add written text to the material.
- Students should be able to access the files and written text.

4.8 Discussion Forums

Description: Every course will have a discussion platform. Students can among themselves or with professors can ask their doubts here. This facilitates better conceptual clarity.
Functional Requirements:

- Every post in the platform should have a title, a phrase that gives a rough idea of what the post is.
- Everyone should be able to do multiple posts if necessary.

4.9 Add and Submit Time based Quizzes

Description:

- Professors can add time based quizzes to their respective courses.
- Students can see this quizzes only when they have started and should submit their responses before the quiz is expired.
- After the quiz is over, the students should be able to see their grade.

Functional Requirements:

- Every quiz will have a start and end time.
- Professors should communicate this to the students beforehand.
- The quiz should only be accessible to the students after the start time of the quiz and should stop accepting responses after the quiz has expired.
- The professor should be able to add as many questions as he wants in the quiz with the facility of making them single correct or multiple correct as seem fit.

5 Other Nonfunctional Requirements

5.1 Performance Requirements

The average response time shall be less than 2 seconds, and further the website should accommodate 1000 booked per minute. In case of a website crash, redundant parts shall resume within 30 seconds and average repair time should be less than 1 hour. Once the server has started, the website must load within 30 seconds. The website should be able to handle upto 100 concurrent users. The database should be able to accommodate no more than ten thousand entries and transactions.

5.2 Security Requirements

First and foremost, the course management system shall run inside a firewall. If no third-party antivirus is found, windows defender firewall shall be used. The system shall support different roles for users, such as Instructors, Students, and administrative staff, the user logged in with given role should only be allowed access consistent with that role. For example, a student shall only be allowed to see he/she grades not to modify it.

5.3 Software Quality Attributes

The system shall not be down more than 2 times a month. The system will be down once a year to update libraries for adaptation to newer versions of python. Scaling the system to large number of users: large courses will have hundreds of students. The system shall be able to handle the load for such courses, especially near assignment deadlines when many students can be expected to access the course management system.

5.4 Business Rules

Professor accounts can only be verified from the order of the university which has bought the system. In any conflict, the word of the developers shall be final. Students will not be allowed to view any database tuples except their own. Sharing your passwords may result in termination of account from the Course Management System.