Software Requirements Specification

SmarTutor

Version 0.1

Prepared by

Group -16 Group Name:Software Avengers

Sonu Kumar	**211052**	**sonuk21@iitk.ac.in**
Ch Hemanth Kumar	**210277**	**chandaka21@iitk.ac.in**
Sarthak Paswan	**220976**	**sarthakp22@iitk.ac.in**
Yash gothwal	**211189**	**yashg21@iitk.ac.in**
Kantule Ritesh Ramdas	**210488**	**kantulerr21@iitk.ac.in**
Saurav Kumar	**210950**	**sauravk21@iitk.ac.in**
Surendra kumar ahirwar	**211083**	**surendrak21@iitk.ac.in**
Rishit Bhutra	**210857**	**rishitb21@iitk.ac.in**
Krishna Chandu	**220832**	**pkrishna22@iitk.ac.in**

course- CS253

Project Mentor - Sumit Chaduhry

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
0.1	Rishit Bhutra Sonu Kumar Ch Hemanth Kumar Sarthak Paswan Yash gothwal Kantule Ritesh Ramdas Saurav Kumar Surendra kumar ahirwar Krishna Chandu	First draft	26/01/24

-
-

1 Introduction

1.1 Product Scope

The product aims to facilitate various needs of students during Academic Pre-Registration of the coming semester at IIT Kanpur. The students can get fundamental information about offered courses and peer reviews about content and instructor of the course. Students can discuss with seniors and themselves in different ways about registering for courses. The application would make it easy for students to resolve timetable clashes during registration. Any modern web browser can access the application.

The main goals and benefits of the application are as follows:

- To provide a platform for the discussions regarding courses during the pre-registration process.
- To resolve timetable clashes easily.
- To help students choose courses based on their reviews and descriptions.

1.2 Intended Audience and Document Overview

1.2.1 Intended Audience

1. Developers: Those who are responsible for coding and proceeding with the implementation of SmarTutor.
2. Project Managers: Those who will calculate the function requirements and oversee the development and progress made in the project.
3. Users: There are two broad categories of the users of the system:
    a) Students: Those who can login and access the product.
    b) Department Mentors: Those who can login and then access and edit the courses accordingly.
4. Testers: Those who can validate the system at the time of testing to verify the completeness and consistency of the system.

1.2.2 Document Overview

First Section: In this section, we give an overview of our product and list its salient characteristics features. This section divides the document into many points to establish the document's overall flow. The reader has all the necessary information at their disposal to comprehend the material in an easy way.

Second Section: In this section, we offer a comprehensive description of the characteristics of our product, taking into account the real-time constraints on its functions as well as the assumptions that were made throughout the project's development. We also provide a brief summary of the different dependencies our product has on different entities.

Third Section: In this section we deal with the specific details of the various components involved in the software. In this section we define the functionalities provided by our product in detail. The customer can verify and examine the consistency of our program by using this part as a benchmark or reference. Using the USE CASE diagram, this part presents a simplified overview of our program by using examples.

Fourth Section: In this section we provide an insight into the non functional requirements that shall be met by our software ranging from appropriate client response to security critical issues. This is especially important for the developers.

Appendices: This section includes the various appendices for the aid of the reader of this SRS document.

1.2.3 Important Sections

1. Developers and Testers can start reading the document with focus on overview (2.1), functionality (2.2), design and implementation (2.3), interfaces (3.1) and functional requirements (3.2).
2. Project Managers should focus on overview (2.1) and functional requirements (3.2) and then the use cases (3.3).
3. Users should focus on the scope (1.1), overview (2.1) and functionality (2.2), and then the specific requirements, going through sections 3.1 and 3.2.

1.3 Definitions, Acronyms and Abbreviations

API: Application Programming Interface

SRS: Software Requirements Specifications

IIT K: Indian Institute for Technology Kanpur

DUGC: Department Undergraduate Committee

SOTA: State of the arts

1.4 Document Conventions

Formatting Conventions:

- This document is written with an Arial font of size 11 with single spacing and 1-inch margins.
- Words highlighted with bold in the same font space represent terms whose explanations are either given in footnotes or separately in the same section.
- Underline has been used for headings in subsections.
- Bullet point and number ordering has been used as a listing typesetting tool.

Naming Conventions:

- Mentors : Department Mentors and DUGC nominees
- Students : Any potential user who are registering for IIT K courses.
- Users : Students and mentors.

1.5 References and Acknowledgments

We'd like to acknowledge the help of our TA, Mr. Sumit Chaduhry, for their valuable input in the creation of this document. We also would like to thank Prof. Indranil Saha for providing the SRS template and teaching the concepts.

2 Overall Description

2.1 Product Overview

For hassle-free Pre-Registration work, Smart Tutor is a one-stop shop. We currently encounter many challenges during the pre-registration period, such as those related to the courses that are being offered and their schedule, and their course overviews, and course timetable conflicts and guidance about a particular course. Therefore, our website will make all of these challenges easier and assist with a challenge free signup.

2.2 Product Functionality

The product should have the following functionality for Department mentors:

- add/ remove a course to the list of courses.
- edit the description of a course.
- publish/ unpublish the courses before pre-registration starts.
- add/ edit Timings, course Instructor to description of course while publishing.
- enroll themselves in the published courses.
- view/ respond/ create the forums of courses that they enrolled in.

The product should have the following functionality for students:

- access only his/ her account.
- search the list of published courses
- view the description of a course
- select a course from the lists of clashing/ non-clashing/ all courses that are published and can view the allotted timings of course.
- view/ respond/ create the forums of the courses anonymously/ publicly that they enrolled in.
- give reviews of the course and the ratings should be displayed in the description of the course.

2.3 Design and Implementation Constraints

- For this application and its dependencies to function properly, we need an operating system with 64-bit architecture and an internet browser such as Google Chrome, Brave, Mozilla Firefox, etc.

- The software's relevancy is mostly dependent on its courses and related materials, such as user databases and quick overviews; hence, the software's memory needs to be adequate.

- The server should have enough resources available to handle and promptly deliver any requests and data made by the user community.

2.4 Assumptions and Dependencies

1. The course mentors can access the courses of their departments only and they are well aware of the courses which will be offered before the actual start of pre- registration.

2. Only IITK communities are the users of our product as some data present on our website are confidential.

3. Our services would be best served on modern web browsers as we are designing with some of the modern web browsers in mind. If someone is using the older versions of web browsers, then the interface might not be efficient. The backend services are hosted on Node.js.

4. Since the internet would play a major role in all of this, it is assumed that the users have a reliable internet connection. The efficient working of the project would depend on the contribution of the technology stack we are using and other services such as mailing servers would affect the time taken for the users to complete a transaction, hence affecting the efficiency of the program. The internet connection would also play some role in all of this, the better the internet connection, the better the things would work.

3 Specific Requirements

3.1 External Interface Requirements

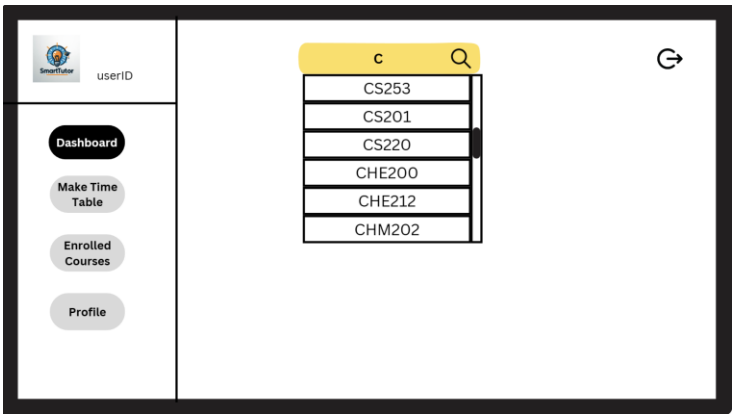
3.1.1 User Interfaces

Sign Up Page: The sign-up window will ask for the IITK email and password of the user. The user can sign up after confirming the password.

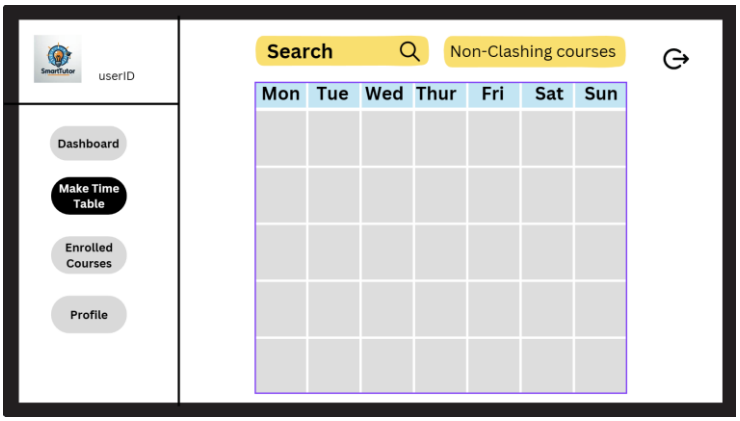
Sign In Page: The sign-in window will ask for the IITK email and password of the user. The user can enter their email and password in the appropriate text fields and click on the Login button to sign in to their account.

Student's Interface: It should display the portfolio, enrolled courses, time table, dashboard.

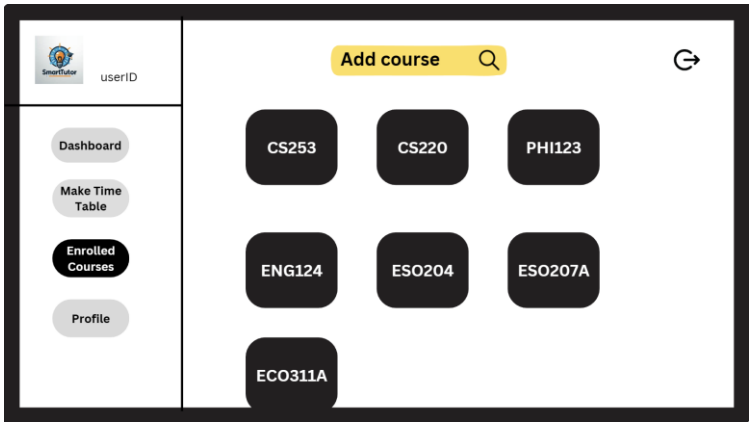
Dashboard: It should display a search bar with all the listed published courses.



Make Time Table: It should display the time table and provide option to add/ remove a course from timetable.

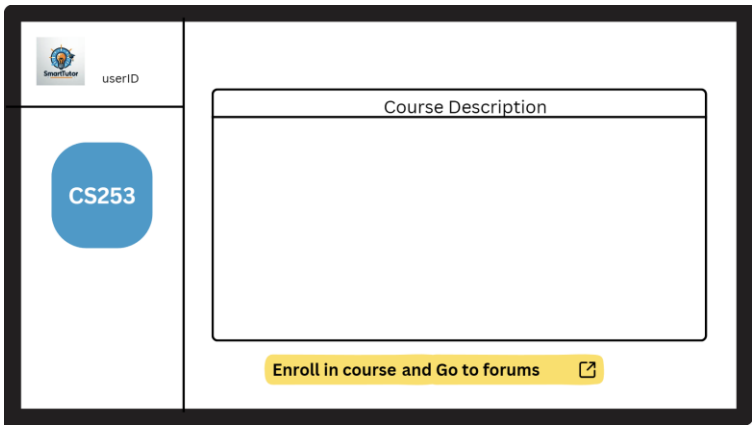


Enrolled Courses: It will show all the courses the student is enrolled in during the semester.

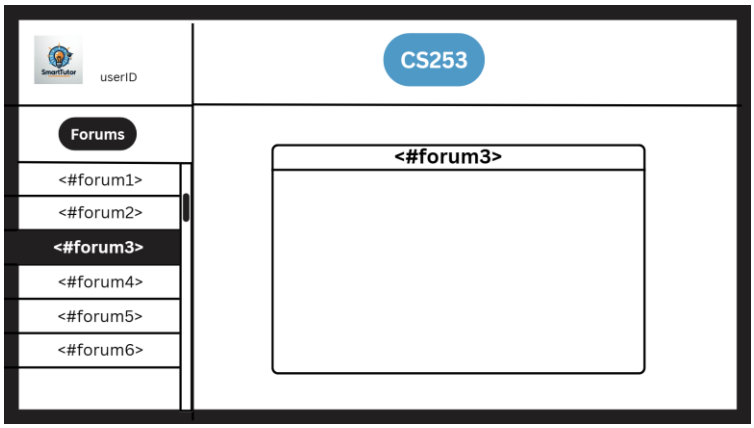


Profile: It will contain the user portfolio.

Course Description Page: The page contains the course title, course description, its structure, instructor information, its policies, schedule, prerequisites, syllabus etc.



Course Forum Page: The page will contain all the forums asked by the students of a particular course. There will be a general discussion area where the queries are asked and the mentors can answer accordingly.



3.1.2 Hardware Interfaces

The system has been designed without any hardware prerequisites. The absence of hardware requirements affords users the flexibility to choose and implement the system on various devices, accommodating individual preferences, budget constraints, and existing infrastructure.

3.1.3 Software Interfaces

There are no software interfaces involved in the project. No third party installations required.

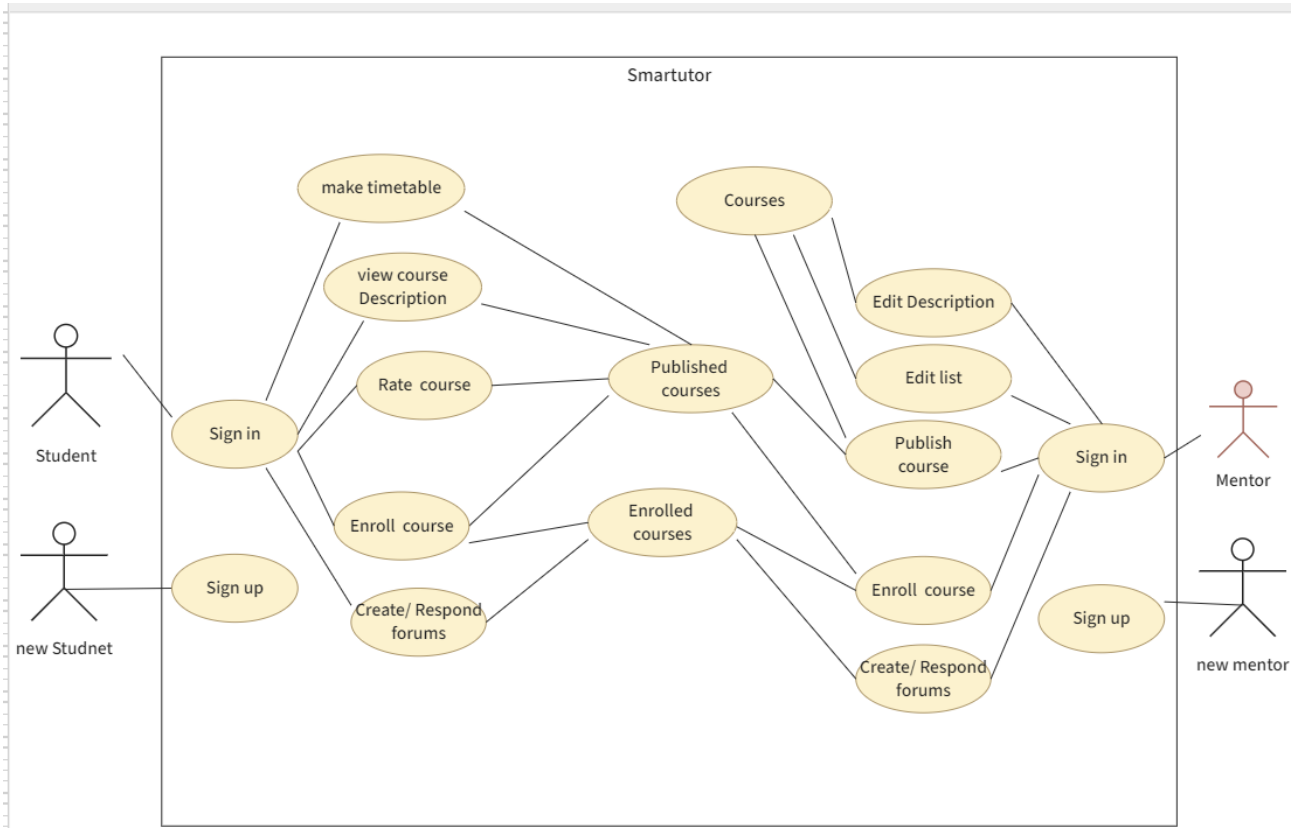
3.2 Functional Requirements

.

1. System shall maintain a list of all courses(offered and unoffered) and description of each course as provided/ updated by Department mentors.
2. Department mentors shall be able to create his/her account.
3. Department mentors shall be able to access his/her account via email and password.
4. Department mentors shall be able to add/ remove a course to the list of courses and shall be able upload description of the course.
5. Department mentors shall be able edit the description of a course.
6. Department mentors shall be able to publish/ unpublish the courses before pre-registration starts.
7. Department mentors shall be able add Timings, course Instructor to description of course while publishing.
8. Department mentors shall be able to enroll themselves in the published courses.
9. Department mentors shall be able to view/ respond/ create the forums of courses that they enrolled in.
10. System shall segregate forums according to whether they are viewed/ unviewed by the Department mentor.
11. Students shall be able to create his/her account.
12. Students shall be able to access his/her account via email and password.
13. Students shall be able to search the list of published courses and view the description of a course.
14. Students shall be able to search and select a course from the lists of clashing/ non-clashing/ all courses that are published and can view the allotted timings of course.
15. Students shall be able to add/ remove the selected course to the Timetable.
16. System shall display the timings and instructor of course while adding/ removing a course.

17. System shall display the added courses in the timetable.
18. Students shall be able to enroll themselves in the selected course.
19. Students shall be able to view/ respond/ create the forums of the courses anonymously/ publicly that they enrolled in.
20. Students shall be able to give reviews of the course and the ratings should be displayed in course description.

3.3 Use Case Model



3.3.1 Use Cases

S.No	#1

Author	Ch Hemanth
Purpose	user registration and account creation
Requirements Traceability	user Name, Email id
Priority	High
Pre- conditions	user name, email id should be correct
Post- conditions	After email verification,user is registered in the database
Actors	User's device
Exceptions	Incorrect email
Includes	None
Notes/ Issues	None

S.No	#2
Author	Ch Hemanth
Purpose	Publishing a course.
Requirements Traceability	list of courses
Priority	High
Pre- conditions	Department mentor must be registered
Post- conditions	course is now added to published courses
Actors	Department mentor's device
Exceptions	none.
Includes	Use case #1
Notes/ Issues	none.

S.No	#3
Author	Ch Hemanth
Purpose	enrolling in a course

Requirements Traceability	User email id, course id
Priority	High
Pre- conditions	Course should be published
Post- conditions	user is now enrolled in the course
Actors	Department mentor's device, Student's device
Exceptions	incorrect Course id
Includes	use case#1, use case #2
Notes/ Issues	Course should be published.

S.No	#4
Author	Ch Hemanth
Purpose	Creating a forum in a course
Requirements Traceability	User's email id, User name(if creating publicly)
Priority	High
Pre- conditions	User should be enrolled in the course
Post- conditions	created forum is added to the list of the forums
Actors	Student's device, Department mentor's device.
Exceptions	none
Includes	use case#1,use case #2, use case #3
Notes/ Issues	user should be enrolled in the course

S.No	#5
Author	Ch Hemanth
Purpose	make timetable by student

Requirements Traceability	list of published courses, Timetable of published courses, student's email id
Priority	High
Pre- conditions	student must be registered and the course must be published
Post- conditions	changes are now updated in students's timetable
Actors	student's device
Exceptions	Incorrect course id.
Includes	Use case #1, Use case#2
Notes/ Issues	none.

S.No	#6
Author	Ch Hemanth
Purpose	Editing course content by Department mentor
Requirements Traceability	list of the courses,course id
Priority	medium
Pre- conditions	Course should be in the list of courses, mentor is registered
Post- conditions	Course description should be updated
Actors	Department mentor's device
Exceptions	Incorrect course id.
Includes	Use case#1
Notes/ Issues	Course should be published again for changes to be displayed in the published course.

4 Other Non-functional Requirements

4.1 Performance Requirements

The interface should be easy to use so that the user and admins can interact with the software in a seamless manner . We can ensure this by providing various functionalities such as drop down menus, search bars, etc.

Performance of the application depends on some other factors like

● APIs: The application is expected to have a traffic of over 5K users. The APIs should be scalable enough to support this traffic. The IITK community is about 20K. We expect 45% of this to be our target audience.
● Concurrency: The database read/write operations should respect the ACID properties as there would be multiple users accessing the same API at the same time so there shouldn't be any concurrency issues
● Latency: It is preferable that API requests take less than 200ms to fulfill, so that the App runs smoothly. This is quite close to the standard expected latency in industry.

4.2 Safety and Security Requirements

● Privacy: The user's email address should never be shared in any way without the user's permission. Same applies for other personally identifiable user data.

● Required login: Prior to utilizing our product, all users are required to create an account. This guarantees users' legitimacy. Security prerequisites.

● Integrity: The software shouldn't taint the history of talks, adds, and drops for mentors and students in the course. It is not acceptable for the mentors' data to be compromised.

● Encryption: SOTA encryption techniques should be used to store any sensitive data, including passwords, in an encrypted format.

• Memory Requirements: To reduce memory and storage needs, the software should make use of current optimization techniques.

4.3 Software Quality Attributes

Availability: All users on campus will have access to and availability of the online application. Any student or department mentor may use this web application without any limitations.

Interoperability: No other application using the device at the same time will be hampered by this one's operations. Users may resume ordering from the Recent screen or browser tabs, use any

other application concurrently, and add the course to the order while adding or removing. This will guarantee that the application is compatible with other systems.

Consumability: The campus junta utilizing the program during pre-reg will find it quite easy to use thanks to its user-friendly interface. Students registered for the course may be viewed on the course, and conversations related to the same subject can be held there as well. The program will name its menus, windows, and choices in a way that makes them mostly self-explanatory.

Maintainability: The system should be designed with as little maintenance overhead as feasible through the architecture, implementation, and software documentation. Software must be designed such that fixing security flaws, which includes testing and updating documentation, can be done in no more than two days by a single person.

Appendix A – Data Dictionary

Will be updated after the outline of the class diagram in the next sprint.

Appendix B - Group Log

Date	Timings	Minutes
10/ 1	8:30pm - 9:30pm	Brainstormed over different project ideas and decided 2 main ideas. 1. hotel management system 2. E- learning platform.
12/ 1	3:00pm - 4:30pm	We discussed the above two ideas as per suggestion given by sir and after intense discussion we decided to go with a platform that helps during Academic pre-registration.- SMARTUTOR
15/ 1	9:00pm - 12:30am	We discussed major functional requirements as per the data we collected from students at IITK about the software.
17/ 1	6:30pm - 7:00pm	Introductory meeting with TA and work delegation in the team to work on several sections of the SRS.
25/ 1	6:30 pm - 7:30pm	Updated TA with our current progress and final discussion on the remaining few sections of SRS.