

Software Requirement Specification (SRS) for Lectures Management System

1.Introduction

1.1 Purpose:

Due to the COVID-19 pandemic in India, our college started online classes. A large number of people began to forget about the lectures, which led to a poor attendance rate.

Despite the fact that online classes could be taken from anywhere in the world, many people found this to be a drawback because they become carefree and neglected to attend the classes in the first place. As a solution to this problem, we developed a reminder system in which our bot sends reminder emails to the provided email address and prompts students to turn in assignments on time.

This product is intended for all those students who must attend classes online but frequently forget to do so.

1.2 Scope:

We describe what features are in the scope of the software and what are not in the scope of the software to be developed.

In Scope:

- a. Lecture reminder mails
- b. Assignment reminder mails
- c. Lecture timing and venue details
- d. Attendance record

Out of Scope:

- a. Mobile app
- b. Whatsapp reminders
- c. Quizzes record
- d. Exams record

1.3 Overview:

The rest of this SRS is organized as follows: Section 2 gives an overall description of the software. It gives what level of proficiency is expected of the user, some general constraints while making the software and some assumptions and dependencies that are assumed. Section 3 gives specific requirements which the software is expected to deliver. Functional requirements are given by various use cases. Some performance requirements and design constraints are also given. Section 4 gives some possible future extensions of the system.

2.Overall Description:

2.1 Product Perspective:

In order to keep track of all the scheduled classes in a day and to remind students to attend classes, Lecture-Bot is designed for students who have a lot of classes or lectures in a day that they hardly ever attend. Our app would give you a one-stop solution to all of your worries if you live in today's world where we spend a lot of time on our mobile devices using social media but still don't have the time to set reminders for our meetings and classes. Once it's running, our bot will notify you via email and also include a link to join the class.

2.2 Product Functions:

Lecture-Bot should support the following use cases:

Class of use cases	Use cases	Description of use cases
Use case related to browsing our app	Open WebApp on browser	<i>Open the home page.</i>
Use cases related to system authorization	Login	<i>Login into lecture-bot</i>
Use cases related to set-up	Choose subject and subscribe	Choose subject and subscribe them for alerts and other features to access
Use cases related to Editing a section	Unsubscribe subjects	<i>Unsubscribe subjects for no alerts and other features</i>
Use cases related to Information display	Display attendance	<i>Displays the attendance of a particular subject</i>
	Display time table	<i>Displays the time table of a particular subject</i>
	Display assignments	<i>Displays all the assignments of a particular subject</i>
	Display pending/done assignments	<i>Displays category wise pending/done assignments of a particular subject qq</i>
	Display all assignments	<i>Displays all the assignments of all the subscribed subjects</i>
Use cases related to computations	Compute attendance	<i>Compute the attendance from the record maintained in the database</i>
Use cases related to alerts	Show pending alerts of a subject	<i>Show all the pending/unseen alerts of a particular subject</i>
	Show all pending alerts	<i>Show all the pending/unseen alerts of all the subscribed subjects</i>
	Delete alerts of a subject	<i>Delete alerts of a particular subject</i>
	Delete all the alerts	<i>Delete all the alerts of all the subscribed subjects</i>
	Class alerts	<i>Sends alerts to attend the scheduled classes</i>
	Assignment alerts	<i>Send alerts to complete the assignment and submit them before the deadline</i>

3. Specific Requirements:

3.1 Functional Requirements:

We describe the functional requirements by giving various use cases.

Use cases related to Authorization:

Use Case 1: Open browser home page

Primary Actor: User

Pre-Condition: Internet connection available.

Main Scenario:

1. User searches Lecture-Bot on the browser.
2. Open Lecture-Bot home page.

Alternate Scenario:

- 2.1. Network failure.

Use Case 2: Login

Primary Actor: User

Pre-Condition: Internet connection available.

Main Scenario:

1. Open login page. (User dashboard is displayed if the user is already logged in)
2. User enters the college email id, semester number and branch.
3. System sends OTP on the entered email id.
4. User enters the OTP.
5. System does authentication.
6. User's dashboard is displayed.

Alternate Scenario :

- 5(a). Authorization fails
- 6(a) 1. Prompt the user if he types the wrong OTP
2. Allow him to re-enter the OTP. (3 chances are given at max)

Use cases related to Set-up:

Use Case 3: Choose subject and subscribe them for alerts and other features to access

Primary Actor: User

Pre-Condition: User logged in.

Main Scenario:

1. User opens the subjects section and selects subjects he wants to subscribe to (As the subjects section will have all the subjects which are available to that particular branch during desired semester) .
2. System opens the section .
3. User then clicks on the subscribe button.
4. User get subscribed to the chosen subjects.

Use cases related to Editing a Section:

Use Case 4: Unsubscribe (for no alerts).

Primary Actor: User.

Pre-Condition: User logged in.

Main Scenario :

1. User opens the subjects section and selects the subjects he wants to unsubscribe .
2. System opens the section.
3. User clicks on the unsubscribe button.
4. User get unsubscribed from the chosen subjects.

Use cases related to information display:

Use Case 5: Display attendance.

Primary Actor: User

Pre-Condition: User logged in.

Main Scenario:

1. User opens the section of the subject from the subjects section for which he wants to know the attendance .
2. System opens the section.

3. User clicks on the show attendance button.
4. System displays the attendance (from the day the user has subscribed for that subject on the basis of how many classes he/she has marked attendance for).

Use Case 6: Display Time Table

Primary Actor: User.

Pre-Condition: User logged in.

Main Scenario:

1. User opens the subject section from the subjects section for which he wants to know the time table.
2. User selects the option of viewing the time table.
3. System displays the time table.

Use Case 7: Display assignments for a particular subject

Primary Actor: User

Pre-Condition: User logged in.

Main Scenario:

1. User opens the subject section from the subjects section for which he wants to display the assignments for.
2. User selects the option of viewing the assignments.
3. System displays all the assignments.

Use Case 8: Display pending/done assignments for a particular subject.

Primary Actor: User

Pre-Condition: User logged in.

Main Scenario:

1. User opens the subject section from the subjects section for which he wants to display the assignments.

2. User selects the option of viewing the assignments .
3. System displays all the assignments.
4. User clicks the pending/done button.
5. System shows the pending/done assignments.

Use Case 9: Display all the assignments of all the subscribed subjects.

Primary Actor: User

Pre-Condition: User logged in.

Main Scenario:

1. User clicks on the assignments button on the user dashboard.
2. System displays assignments for all the subscribed subjects.

Use cases related to computations:

Use Case 10: Compute attendance.

Primary Actor: System.

Pre-Condition: User logged in.

Main Scenario :

1. System computes the attendance with the help of a record.

Use cases related to alerts:

Use Case 11: Show the pending alerts of the specific subjects.

Primary Actor: User.

Pre-Condition: User logged in.

Main Scenario :

1. User goes to the main page.
2. User initiates the “show pending alerts” functionality.
3. System shows the pending alerts regarding the assignments or other particularities.
4. User scrolls the scroll bar and chooses the subject he wants to see alerts for.

Use Case 12: Show all the pending alerts.

Primary Actor: User.

Pre-Condition: User logged in.

Main Scenario :

1. User goes to the dashboard.
2. User initiates the “show pending alerts” functionality.
3. System shows all the pending alerts of all the subscribed subjects.

Use Case 13: Delete alerts of specific subjects.

Primary Actor: User.

Pre-Condition: User logged in.

Main Scenario :

1. User goes to the dashboard.
2. User initiates the “delete alerts” functionality.
3. User clicks on that particular subject section for which he wants to delete alerts for.
4. User clicks on the alert he wants to delete (Because he may have already done that work).
5. User is asked for confirmation.
6. The alert is deleted.

Use Case 14: Delete all alerts.

Primary Actor: User.

Pre-Condition: User logged in.

Main Scenario :

1. User goes to the dashboard.
2. User initiates the “delete alerts” functionality.
3. User selects the “all alerts” option.
4. User clicks delete alerts button.
5. User is asked for confirmation.
6. All the alerts are then deleted.

Use Case 15: Send class alert via email.

Primary Actor: System.

Pre-Condition: User logged in, Internet connection available.

Main Scenario :

1. System sends a class alert to the registered email id 30 minutes prior to the class.
2. Alert is sent every 5 minutes until the user clicks on the "IN CLASS" (taken as present for record) or "NOT ATTENDING" option (taken as absent for record).
3. No alert is sent after 30 minutes of class start time.
4. If no response is initiated by the user, then after 30 minutes from the start of class, user will be marked as absent.

Use Case 16: Send assignment alerts.

Primary Actor: System.

Pre-Condition: User logged in, Internet connection available.

Main Scenario :

1. System sends reminders to the user 48 hr, 24hr, 12 hr, 6hr, 3hr, 1hr and 10 minutes before the assignment deadline via email.

3.2 **Hardware interfaces:**

Only the recommended configuration (basic requirements) needed. No specific hardware required.

3.3 **Software interfaces:**

1. Browser to load and view web pages.
2. Operating system.

3.4 **Design Constraints:**

1. **Security:** The files containing important information about the college and the user should be secured against malicious deformations.
2. **Fault Tolerance:** Data should be saved automatically and prevented from being corrupted in case of system crash or power failure or network failure.

4.Future Extensions:

As we have made a web app, a future extension could be converting this web app to mobile app as it is more convenient to access by users. Along with lecture and assignment reminders, we can also add the functionality of quizzes as well as exams reminders along with their complete details. Now, as WhatsApp is very popular among students, we can shift the reminders on mail to WhatsApp using the WhatsApp API (Application programming Interface).