Software Requirement Specification for Seminar and Talks Management System

Team 33

Jatin Tarachandani
CS20BTECH11021

Prashanth Sriram S
CS20BTECH11039

Shambhu Kavir
CS20BTECH11045

Taha Adeel Mohammed
CS20BTECH11052

February 25, 2023

Contents

1 Introduction

1.1 Definitions and Acronyms

- SeTMaS: Seminar and Talks Management System, the name of the application
- IITH: Indian Institute of Technology Hyderabad
- Talk: Any Seminar or Webinar or Event that needs to be booked on a particular venue and for a particular time duration.
- Super Admin: An admin who is granted elevated privileges over the other admins of the system. These privileges include adding and removing admin privileges from other users, transferring his status as super admin to another admin. Addition and removal of other admins can only be done by the super admin.

1.2 Purpose of this application

This application is a seminar/talk booking management system: where faculty and students alike can place requests to book certain rooms at certain times to conduct talks and seminars, and a centralized platform for the administration governing these bookings to view requests, and accept and reject them as appropriate. This application is also proposed to be integrated into the IITH mailing system, as described in the appendix below.

This requirement specification is meant to serve as a guideline for what functionality is supported within this project; and what use cases are fulfilled by the app.

1.3 Scope of the application

The functionalities fulfilled by this application are:

- Faculty/students/other IITH people will be able to submit requests to book one of the available rooms for a specific length of time as per their convenience. These requests will be approved by a governing authority consisting of approved administrators.
- Said approved administrators will be able to view all the pending requests for booking, and be able to accept and reject them based on the content, time and date of the requester.
- Even without login, any user will be able to view a calendar which displays all the upcoming seminars/talks along with their details.
- This system will also be integrated with the IITH mailing system, with emails being sent to concerned parties when requests are made, accepted, and rejected. This system is elaborated on in Appendix 4.1.

1.4 Overview of this SRS

The next few sections of this SRS talk about the use cases of the application. These range from things like installation, to making a request for a seminar, to the admin functionality of approving or denying requests for seminars.
Section 2 discusses the various users of the application, and tabulates all the use cases which are supported by the application. There are also some assumptions and constraints taken into account when planning the requirements for this software, which are also specified in section 2.
Section 3 is an elaboration on every use case tabulated in section 2. It would consist of details like primary actors, preconditions, and main and alternate scenarios. Design constraints and processes that we thought of are also specified in this section. The appendices in section 4 deal with the UI and the IITH mailing integration.

2 Description

2.1 Product Perspective

SeTMaS is aimed towards a person who wants to book seminars as well as view the schedule of the seminars in the IITH campus. SeTMaS is intended to be a web-app that will run on any modern browser on any device. SeTMaS should be user-friendly, intuitive to use and be a reliable application.

2.2 Product Functions

The use cases supported by SeTMaS are shown in Table 1.

2.3 Principal Actors

1. **Guests:** be able to view the public calendar on the web-app
2. **Users:** be able to log-in and book seminars/talks
3. **Admin:** be able to Log-in and Approve/Reject booking requests.
4. **Superadmin:** be able to add or remove admins, may transfer the role of superadmin to another admin. There can only be one superadmin recognised by the system at a given time.
5. **system:** manages the back-end computations and sends out mail notifications as appropriate

2.4 Assumptions and Constraints

1. The full working of SeTMaS depends on a working internet connection and the running of the back-end server of SeTMaS.

Table 1: Product Functions: Use Cases

Class of use cases	**Use cases**	**Description of use cases**
Use cases related to Authorization	Login	Login into SeTMaS using google sign in with IITH id
Use Cases related to Publishing Info	Display Calendar	Display the calendar with all the accepted seminars
Use Cases related to Booking a Slot	Request Booking	Users can request a room for a time duration
	View My Bookings	Users can view all the booking requests they have made and see the status of them
	Cancel Pending Request	Users can cancel their own pending request
	View Booking Requests	A logged-in valid admin can view all the booking requests made by users in a user-friendly view
	Approve Booking Request	A logged-in valid admin can approve a booking request raised by a user
	Reject Booking Request	A logged-in valid admin can reject a booking request raised by a user
	Cancel Approved Request	A logged-in valid admin can cancel an approved request
Use cases related to Admin Management	View Admins list	Any admin can view the list of all the existing admins
	Add an Admin	The super-admin can add a user to the admin list
	Remove an Admin	The super-admin can remove an admin from the admin list
	Transfer Super-Admin role	The super-admin can transfer their role to another admin
Use cases related to Mailing Notifications	Booking request confirmation	A mail should be sent to the user confirming that the request has been made
	Booking Approval notification	A mail should be sent to the user conveying that the booking is approved
	Booking Rejection notification	A mail should be sent to the user conveying that the booking is rejected
	Seminar Notification	Mails should be sent to those subscribed notifying them of the booked seminar
	Seminar Reminder	Mails should be sent to those subscribed reminding them of the upcoming seminars
	Seminar Cancellation Notification	Mails should be sent to those subscribed notifying them of the cancellation

2. All the users will have a valid IITH email id to access the system.
3. All the users will have access to a modern browser that can run JavaScript.
4. For an administrator to carry out administrative functions in the application, his IITH email ID must be registered as an admin by the super admin of the system. The details of this process are described in this document.

3 Specific Requirements and Use Cases

Here we describe the functional requirements of our application by elaborating on the various use cases given in the previous section.

3.1 Authorization Use Cases

Authorization/Login:

- *Primary Actor*: User, or Admin, or Superadmin (hereafter in this use case referred to as 'client')
- *Precondition*: The client has access to their IITH Google account, and has a working connection to the internet.
- *Main Scenario*:
  1. Client connects to the web application using a compatible browser.
  2. Client clicks on the login button specified on the home page of the application.
  3. Client is prompted to login using their Google account. The Google accounts shown to the client will be the ones they have logged in to on that browser, not necessarily just their IITH Google account.
  4. Client logs in through Google using their required IITH account.
  5. The main authorized user screen of the application is displayed.
- *Alternate Scenario*:
  1. Authentication fails. In this case, prompt the user that they have not used a valid IITH Google account to login, and provide a button to go back to the guest homepage.

3.2 Publishing Info Use Case

Display Calendar:

- *Primary Actor*: Any kind of client (Guest, User, Admin, Superadmin), needs no further credentials

- *Precondition*: The client has a working connection to the internet.
- *Main Scenario*:
  1. Client connects to the web application using a compatible browser.
  2. The calendar with all the upcoming seminars along with their details is displayed on the homepage.

3.3 Booking Slots Use Cases

Request Booking

- *Primary Actor*: Most often a normal authenticated user, however this functionality can be used by admins or superadmin.
- *Precondition*: The user should be logged in and authenticated using their IITH Google Account via the login process explained above.
- *Main Scenario*:
  1. Client initiates the slot booking process.
  2. Client selects a date from the available selections.
  3. Client is prompted to enter the details of the seminar. This includes
     (a) Title
     (b) Details
     (c) The starting and ending time of the room booking
     (d) The department
     (e) The email ID of the requester
     (f) List of the mailing lists, in which the notification and reminders are to be sent. Such as: seminar@cse.iith.ac.in, etc.
     (g) List of time durations, before which reminder mails must be sent. Such as 30 minutes before, 24 hours before, etc.
  4. Assuming the aforementioned details are entered correctly, the request will be submitted.
  5. If the system registers the request, A mail confirming the same is sent to the client.
- *Alternate Scenario*:
  1. User enters an incorrect time (i.e.) a time that overlaps with a **confirmed** booking(this will be visible when selecting a date). In this case, an error popup will be seen when trying to create the request for room booking, and it will not go through to the system.

2. User enters an incorrect or infeasible time duration for the reminder, for example, greater than 72* hours, or when the time at which the reminder is to be sent precedes the time of the booking request. In this case, an error popup will be seen notifying the user of the same to correct it and it will not register as a successful request
3. User enters too many time durations for the reminder, (greater than 6*) in which case, an error popup will be seen notifying the user of the same to correct it and it will not register as a successful request. This is done to prevent the spamming of reminder mails.

View My Bookings

- *Primary Actor*: A user, admin, or superadmin, hereafter referred to in this case as client.
- *Precondition*: Must have successfully logged in to the application. Ideally, they must also have made at least one request(either pending, accepted, or rejected) to have this functionality be meaningful.
- *Main Scenario*:
  1. The client initiates the "View My Bookings" functionality.
  2. The client can then view a list of their created bookings requests, and an indication for each one depending on whether it was accepted, rejected, or is still pending approval by authority. If the client has not yet made any booking requests, then the subpage will still be accessible, however there will be no content on it.
  3. In case one of the requests is pending approval, the user also has an option to cancel the request. Refer the 'Cancel Pending Request' use case.

Cancel Pending Request

- *Primary Actor*: A user, admin, or superadmin, hereafter referred to in this case as client.
- *Precondition*: Must have successfully logged in to the application and is in the View My Bookings page. Ideally, they must also have made at least one pending request to have this functionality be meaningful.
- *Main Scenario*:
  1. The client initiates the "Cancel Pending Request" functionality for a pending request.
  2. The client can then confirm that they indeed mean to cancel the pending request in a pop-up that appears asking if they want to cancel that request, in which case, the request gets cancelled and will not appear anymore as a pending request for any admin to possibly approve.

- *Alternate Scenario*:

    1. The client does not click confirm in the confirmation pop-up, in which case, the request is not cancelled and the client is back in the 'View My Bookings' page.
    2. Before the client's confirmation of the cancellation is processed, an admin had approved the same request, in which case, the cancellation fails since the request was just approved by the admin. In this case, a pop-up appears informing the client of the same and lands the client back in the 'View My Bookings' page. In the unlikely event this happens, the client is advised to contact an admin, since only admins can cancel an approved request. See 'Cancel Approved Request' use case.

View Booking Requests

- *Primary Actor*: A user with admin privileges i.e an admin or a superadmin.
- *Precondition*: Must have successfully logged into the application as an admin. Another condition is that some booking request by some user must have been made, or this use case is not meaningful.
- *Main Scenario*:

    1. The user initiates the "View Booking Requests" Functionality.
    2. The user can see the pending booking requests. From this screen, they can take action to approve or reject any of the pending requests.

Approve Booking Requests

- *Primary Actor*: A verified admin or superadmin.
- *Precondition*: The admin or superadmin has successfully logged in to the application.
- *Main Scenario*:

    1. The admin will be able to view a list of all the pending booking requests.
    2. The admin will select a request.
    3. The admin will be able to view the salient features of the highlighted request. These include name of the requester, date, time, and details of the seminar/talk.
    4. In this aforementioned view, the admin will be able to use the "Accept Booking" functionality.
    5. Immediately after approval, the Booking Approval Notification mail is sent

6. Also, the Seminar Notification mail is also sent
7. A time-trigger is created in the system at the required times as requested in the booking request for sending reminder mails. This triggers will make the system send the reminder emails to those subscribed.

- *Alternate Scenarios*:

1. In case the admin attempts to approve a request which is clashing with an already approved request(this may have been done by either him or a different admin), then an error will be indicated saying that the slots are clashing. After the popup is dismissed, the admin will return to the same view of the details of the requests as in step 3 of the main scenario.

Reject Booking Requests

- *Primary Actor*: A verified admin or superadmin.
- *Precondition*: The admin or superadmin has successfully logged in to the application.
- *Main Scenario*:

1. The admin will be able to view a list of all the pending booking requests.
2. The admin will select a request.
3. The admin will be able to view the salient features of the highlighted request. These include name of the requester, date, time, and details of the seminar/talk.
4. In this aforementioned view, the admin will be able to use the "Reject Booking" functionality.
5. On rejection of a request, A mail informing of the rejection is sent to the requester.

Cancel Approved Request

- *Primary Actor*: An admin or superadmin, hereby referred to as client
- *Precondition*: Must have successfully logged in to the application. Ideally, at least one approved request should be present for this functionality to be meaningful.
- *Main Scenario*:

1. The client initiates the "Cancel Approved Requests" functionality
2. A list of Approved Requests appears on the page. The client can select one of them and initiate the cancel request functionality.

3. Ideally, the client uses this functionality only when there is a legitimate reason to cancel the booking, such as a request from the user who booked the request, or during some extra-ordinary circumstance.
4. The client can then confirm that they indeed mean to cancel the approved request in a pop-up that appears asking if they want to cancel that request, in which case, the booking gets cancelled and the corresponding time duration is now again available.
5. Additionally, A Seminar Notification mail is sent to those subscribed informing them of the cancellation of the event.
6. Also, the time triggers set for the reminder for this event which was set at the time of approval of the request, are now removed from the system.

- *Alternate Scenario*:

  1. The client does not click confirm in the confirmation pop-up, in which case, the request is not cancelled and the client is back in the 'Cancel Approved Bookings' page.

3.4 Admin Management Use Cases

View Admins list:

- *Primary Actor*: Admins and super admins, referred to as clients.
- *Precondition*: The client should be logged in and authenticated using their IITH Google Account.
- *Main Scenario*:

  1. Client accesses the Admin list page.
  2. A list with all the existing admins is displayed to the client.
  3. The super admin will be highlighted in the list.

- *Alternate Scenario*:

  1. If the client was previously able to access the admin list but is now unable to, this means that they have been dismissed as an admin by the superadmin. The client must then contact the superadmin if he desires further clarification.

Add an Admin:

- *Primary Actor*: The super admin, referred to as client.
- *Precondition*: The client should be logged in and authenticated using their IITH Google Account.
- *Main Scenario*:

1. Client accesses the Admins list.
2. Client selects the option to add a new admin.
3. Client enters the new admin's email-ID to add them.
4. A pop up appears asking if the client indeed wants to add this new admin which the client confirms, and a new admin is added.

- *Alternate Scenario*:

1. The client rejects the confirmation pop up, in which case a new admin is not added.
2. The client tries to add an admin who is non-existent, i.e. makes an error in entering a valid email-ID, In this case, an error pops up saying the email-ID entered is invalid. The client is then prompted to enter a valid email-ID.
3. The client tries to add an already existing admin, in which case an error pops up saying that the email-ID entered already has admin privileges. The client is then prompted to enter a valid email-ID.
4. The client tries to add a non-IITH email-ID, in which case an error pops up saying that only IITH email-IDs can be added. The client is then prompted to enter a valid email-ID.

Remove an Admin:

- *Primary Actor*: The super admin, referred to as client.
- *Precondition*: The client should be logged in and authenticated using their IITH Google Account.
- *Main Scenario*:

1. Client accesses the Admins list.
2. To remove a particular admin, the client selects the 'remove as admin' option present alongside the userID of the admin.
3. A confirmation pop up appears asking whether the client indeed wants to remove this particular admin, which the client confirms and the admin is removed.

- *Alternate Scenario*:

1. The client rejects the confirmation pop up, in which case the admin is not removed.

Transfer Super Admin role:

- *Primary Actor*: The super admin, referred to as client.
- *Precondition*: The client should be logged in and authenticated using their IITH Google Account.

- *Main Scenario*:
  1. Client accesses the Admins list.
  2. To transfer the super admin powers to a particular admin, the client selects the 'make super admin' option present alongside the userID of the admin
  3. A confirmation pop up appears asking the client whether the client indeed wants to make this admin the super admin, which the client confirms.
  4. On doing this, client loses their privileges as super admin and becomes an ordinary admin.
  5. The ordinary admin selected by the previous super admin now possesses the system privileges of a super admin.
- *Alternate Scenario*:
  1. The client rejects the confirmation pop up, in which case the client retains the system privileges of a super admin.

3.5 Mailing Notification Use Cases

Booking Request Confirmation

- *Primary Actor*: System.
- *Precondition*: A successful booking request was made by an authorized user (need not be an admin).
- *Main Scenario*:
  1. The system received the successful booking request.
  2. The system uses the fields in the booking request such as email ID of the requester, time, date, etc. to generate an email to the requester informing them that their request has been submitted.
  3. This mail is then sent to the requester's IITH email ID. It is not intended that the requester reply to this mail or any of the autogenerated mails mentioned in this SRS. They are instead encouraged to talk to an admin through another channel like a direct email if they have any queries.

Booking Approval Notification

- *Primary Actor*: System.
- *Precondition*: An admin or superadmin approved a booking request made by a user.
- *Main Scenario*:

1. The system received the successful acceptance of the booking request.
2. The system uses the fields in the booking request such as email ID of the requester, time, date, email ID of the admin, etc. to generate an email to the requester informing them that their request has been accepted.
3. This mail is then sent to the requester's IITH email ID. It is not intended that the requester reply to this mail or any of the auto-generated mails mentioned in this SRS. They are instead encouraged to talk to an admin through another channel like a direct email if they have any queries.

Booking Rejection Notification

- *Primary Actor*: System.
- *Precondition*: An admin or superadmin rejected a booking request made by a user.
- *Main Scenario*:

  1. The system received the successful rejection of the booking request.
  2. The system uses the fields in the booking request such as email ID of the requester, time, date, email ID of the admin, etc. to generate an email to the requester informing them that their request has been rejected.
  3. This mail is then sent to the requester's IITH email ID. It is not intended that the requester reply to this mail or any of the autogenerated mails mentioned in this SRS. They are instead encouraged to talk to an admin through another channel like a direct email if they have any queries.

Seminar Notification

- *Primary Actor*: System.
- *Precondition*: An admin or superadmin approved a booking request made by a user.
- *Main Scenario*:

  1. The system received the successful acceptance of the booking request.
  2. The system uses the fields in the booking request such as email ID of the requester, time, date, email ID of the admin, etc. to generate an email that can notify people of the occurrence of the event.
  3. The system uses the list of mailing lists selected during the time of booking as the recipients of this mail.

4. It is not intended for the recipients of this mail to directly reply to this mail, but they are rather encouraged to send an email to the seminar convener mentioned in the mail or talk to an admin, appropriately depending on the type of their query, if any exist.

Seminar Reminder

- *Primary Actor*: System.
- *Precondition*: A time trigger for a reminder mail is defined by the seminar requester who made the booking.
- *Main Scenario*:
    1. When this is triggered, the system sends out a mail to everyone in the mailing lists selected at the time of the booking associated with the reminder trigger, reminding them of the seminar along with the necessary details like title, abstract, venue, and time.
    2. It is not intended for the recipients of this mail to directly reply to this mail, but they are rather encouraged to contact the email of the seminar convener mentioned in the mail or talk to an admin, appropriately depending on the type of their query, if any exist.

Seminar Cancellation Notification

- *Primary Actor*: System.
- *Precondition*: An admin or superadmin cancelled an approved booking of a seminar
- *Main Scenario*:
    1. The system received the successful cancellation of the booking.
    2. The system uses the fields in the booking request such as email ID of the requester, time, date, email ID of the admin, etc. to generate an email that can notify people of the cancellation of the event.
    3. The system uses the list of mailing lists selected during the time of booking as the recipients of this mail and send the mail.
    4. It is not intended for the recipients of this mail to directly reply to this mail, but they are rather encouraged to contact the email of the seminar convener mentioned in the mail or talk to an admin, appropriately depending on the type of their query, if any exist.

4 Appendix

4.1 Mailing System Integration

Our software is proposed to be integrated with the IITH email ecosystem. This would streamline tasks such as sending of seminar reminders, acceptance/rejection of booking requests, etc.

1. **Superadmin email id**: For best results and clarity, we propose that the super admin of this application be a specially created email address (along the lines of setmas.admin@iith.ac.in for example). This is done so that no one personal user has complete power over the application. In an ideal scenario, this account permanently remains the super admin of this software and the other members of admin who are necessary for managing seminar bookings are added and dismissed by this account as necessary. The transfer use case is only kept in the case of redundancy.

2. **A no-reply email account** is proposed, and this email account will be the sender for all the emails pertaining to SeTMaS operations. This includes:

   (a) Sending of confirmation emails to the requester on successful submission, acceptance and rejection of their seminar/talk booking requests

   (b) Sending Notifying/Reminder emails to groups and mailing lists like seminar@cse.iith.ac.in, seminar@mae.iith.ac.in etc, based on the booking requester's choice of mailing lists to inform.

   (c) It is not intended that anyone reply to this email account, since this email will exist only for the purpose of automated sending of emails. The involved party (such as the convener of the seminar) is planned to be mentioned either through CC or through a field in the mail body, in case someone wants to reply to the mail.

   Specifically, the proposed email ID is along the lines of setmas-no-reply@iith.ac.in, for example.

4.2 UI Design

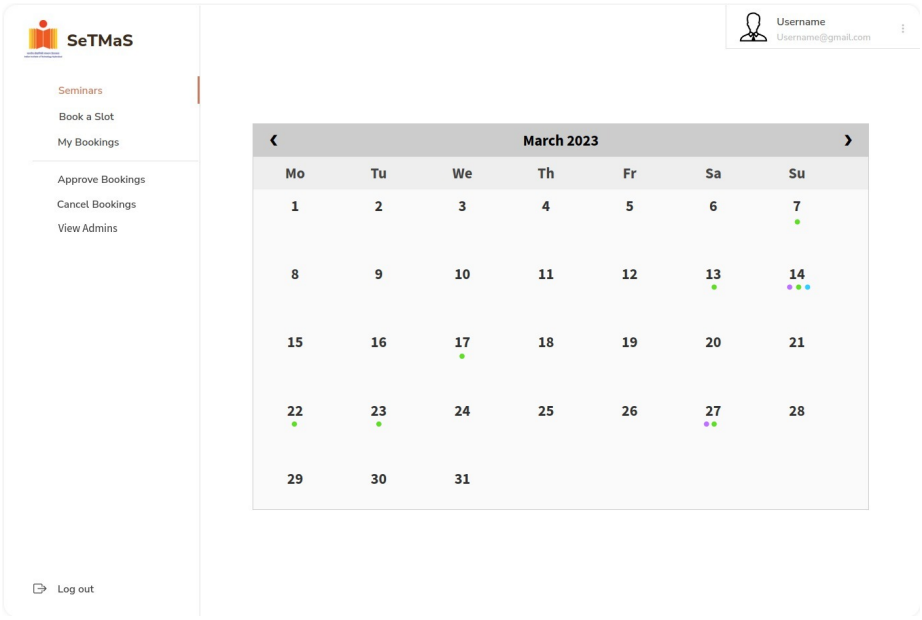

Figure 1: Admin Home Page

Above is a potential UI sketch of the home page for a logged-in admin. Clicking on a date opens another view with the list of bookings for that day. Clicking on a booking opens a view with the details of the booking. Similarly