Software requirement specification document for project What's Next ?

Codecrafters
Supervised by: Dr. Indranil Saha

January 27, 2023

Varun Tokas	211152	varuntokas21@iitk.ac.in
Paras Sikarwar	210699	parassikarwar204@gmail.com
Kruthi Akkinepally	210088	akruthi21@iitk.ac.in
Geetika Gupta	210392	geetika@iitk.ac.in
Apoorva Gupta	210179	apoorvag21@iitk.ac.in
Aditya Kumar	210060	adityakum21@iitk.ac.in
Chitwan Goel	210295	chitwang21@iitk.ac.in
Talin Gupta	211095	taling21@iitk.ac.in
Krish Sharma	210530	krish21@iitk.ac.in
Siddharth Kalra	211032	siddharthk21@iitk.ac.in
Aman Arya	210106	aarya21@iitk.ac.in

Course : CS253

Mentor TA : Archi Gupta

Contents

1 Introduction

1.1 Product Scope

What's Next is a web-based application that equips the student community of IITK with real-time information about various events happening at IITK. It provides a wide array of functionalities such as announcements of various events conducted by the Students' Gymkhana, live scheduling of lectures happening in the LHC. It would also enable students to view events of their particular interests. The application serves as an all-in-one platform to acquaint the user with the daily affairs of IITK, by providing them a feature to organize their endeavors accordingly and derive maximum benefit from it.

For the organizers of events happening in IITK , it would help them in informing the campus community about their events. Detailed information like location, description of event will be provided which would act a form of attracting audience to these events. It would also help in preventing clashes with other events by giving appropriate warnings.

1.2 Intended audience and Document Overview

1.2.1 Intended audience

1. **DEVELOPERS** : For system developers , the most relevant sections include
   (a) Product functionalities
   (b) Design and Implementation Constraints
   (c) External Interface Requirements
   (d) Performance Requirements
   (e) Safety and Security Requirements
   (f) Functional Requirements
2. **PROJECT MANAGERS** :For project managers, the most relevant sections include :
   (a) **Product Functionality** : By evaluating the software's functionalities at a given point in time, the project managers can check how much progress is made in the project.
   (b) **Performance Requirements** : By referring to this section, the managers can evaluate the software in terms of how much performance is delivered by the software, like whether it meets the time requirements such as delay and processing time up to a certain standard.
3. **TESTERS** :For testors , the most relevant sections include **use case models** so that they can test the software in specific scenarios to point out vulnerabilities in the software

1.2.2 Document Overview

1. **Section1** : This sections provides an introduction to our product highlighting it's key features in a verbatim manner. This section defines the overall flow of the document by segmenting the whole document into various pointers so that the reader is equipped with any sort of knowledge he or she may or should posses to understand the document in an affluent manner.

2. **Section2** : In this section we try to entail an in-depth explanation of the features of our product, while mentioning the real time constraints on the functionalities provided by the product and the assumptions made during the development of the project. We also give a brief initiation of the various dependencies that our product have on various entities .

3. **Section3** : In this section we deal with the specific details of the various components involved in the software , ranging from hardware interfaces to software interfaces.In this section we define the functionalities provided by our product in detail . The client can use this section as a reference or benchmark to check and validate the consistency of our software. This section provides a simplified view of our software by drawing instances through the aid of USE CASE.

4. **Section4** : 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 specially important for the developers.

5. **Section5** : This section includes the various appendices for the aid of the reader of this SRS document.

1.3 Definitions, Acronyms and Abbreviations

1. **SUPER ADMIN** : It refers to the developer or team of developers who is responsible for the maintenance of the website and processing day to day requests put up the stakeholders/users of the website.

2. **ADMIN** : This includes people at positions of responsibility under the banner of Students' Gymkhana from top to bottom, ranging from President to Club Coordinators or Team Heads.

3. **SPECIAL EVENTS** : Some events which are organzied on a large scale such as college festivals, research syposiums , panel talks by reputed personalities have more importance as compared to small scale events organized by college clubs and societies . For example : an event in **Techkriti** is special while a literary discussion by ELS is not. So events with bigger scale and greater importance are refered to as special events.

4. **LHC** : Lecture Hall Complex

5. **API** : API : Application Programming Interface

6. **GUI** : GUI : Graphical User Interface

1.4 Document Conventions

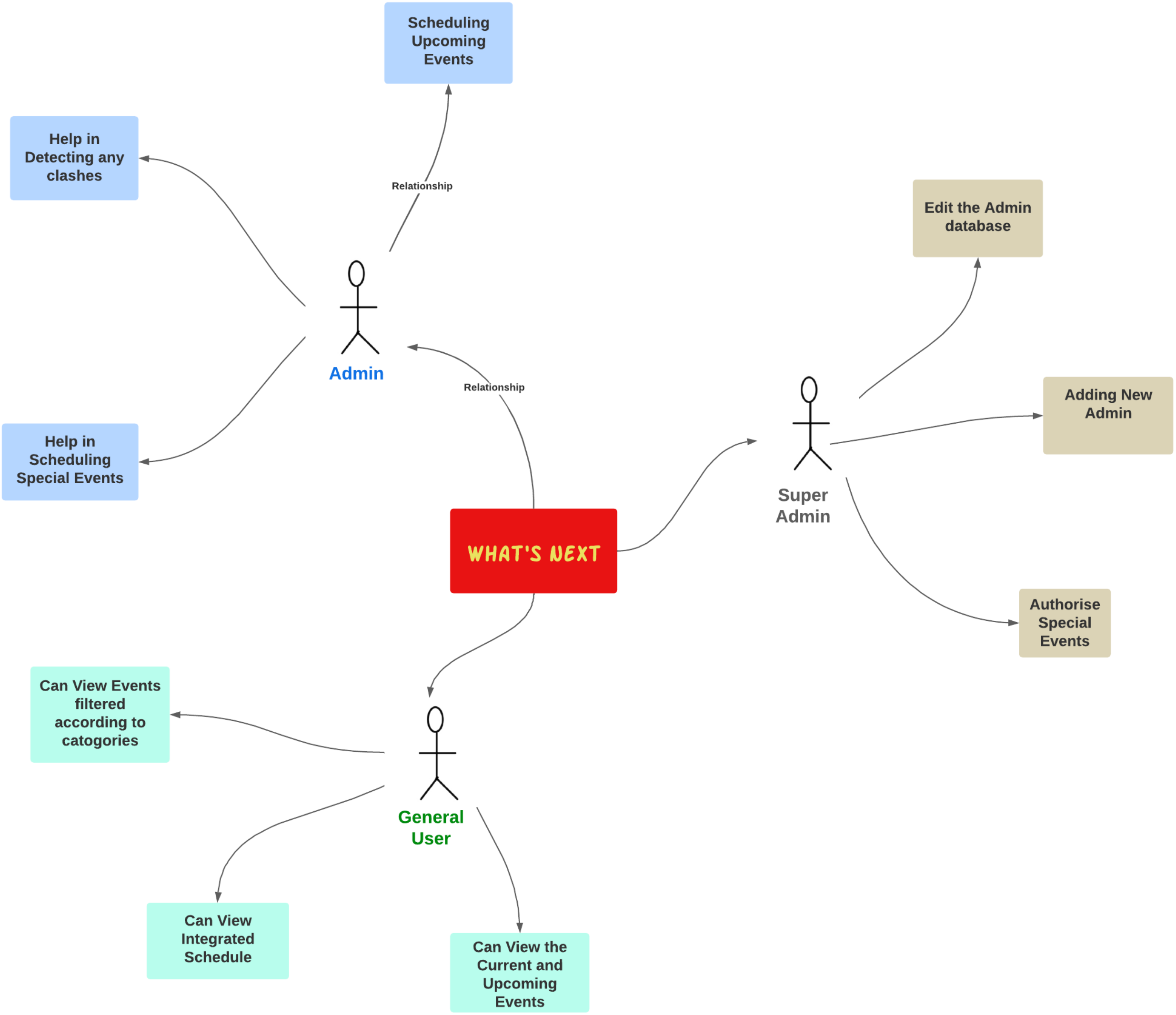
While preparing this document, to make readability user-friendly and important parts clear, we have used following conventions:

- The headings of all sections are written in **bold**.
- Headings of subsections are **bold** and written in the font size 14.
- The content in the sections is written in Arial font size 11.
- Any important term or short term is written in **bold**.
- The alignment of whole content is **justify**.
- Text has been indented wherever required to highlight the hierarchy of the content.
- The document follows the latex formatting, indenting and the numbering conventions. Any derivations from the same will be explicitly specified.

1.5 References and Acknowledgements

Style guide (for this document):
IEEE Software Requirements Specification Template.
Reference SRS document:
Reference

2 Overall Description


Scheduling Upcoming Events
Help in Detecting any clashes
Relationship
Edit the Admin database
Admin
Relationship
Adding New Admin
Help in Scheduling Special Events
Super Admin
WHAT'S NEXT
Authorise Special Events
Can View Events filtered according to catogories
General User
Can View Integrated Schedule
Can View the Current and Upcoming Events

OVERALL DESCRIPTION DIAGRAM

2.1 Product Overview

What's Next is a one-stop solution for all that is going on in IITK. Right now, IITK Janta gets many emails about what is happening on campus. Clubs, sports, academics, etc., are just a few. Among so many unimportant emails, students often miss out on the emails that matter to them. Sometimes students are free and want to know what is happening on campus so they can visit the venue and have fun with their friends.

To help the campus Junta to know what is going on right now and to sort all the activities according to their interests, we have our product. Students will get information about all events and lectures in one centralised location sorted according to their interests and can use their free time better.

The clubs now send emails to the students when they schedule their club activities. But these events often clash with events from other clubs. Also, it's difficult for clubs to send emails for last-minute timing changes. Our product makes it seamless for the clubs to add their events and notifies them of any possible clashes. They can also add posters, locations and other essential details to help the students.

2.2 Product Functionality

- The system would provide an easy interface to the admins for scheduling upcoming events.
- The system would help the admins in detecting any clash.
- The system would allow the admin to schedule their event as special event.
- The user would be able to view the current and upcoming events sorted according to their scheduled time.
- The system would enable the users to view events filtered according to categories.
- The system would display an integrated schedule
- Super Admin will be able to create data fields for the admin login credentials and will also be able to allow an admin to schedule a special event.

2.3 Design and Implementation Constraints

- Events database forms an integral part of the relevance of the software, its memory should be sufficient.
- Server should have sufficient resources available in order to accommodate and serve all the requests and the data requested by the users community, on time.
- Grant of making an event special might require time as it is hands of super-admin

2.4 Assumptions and Dependencies

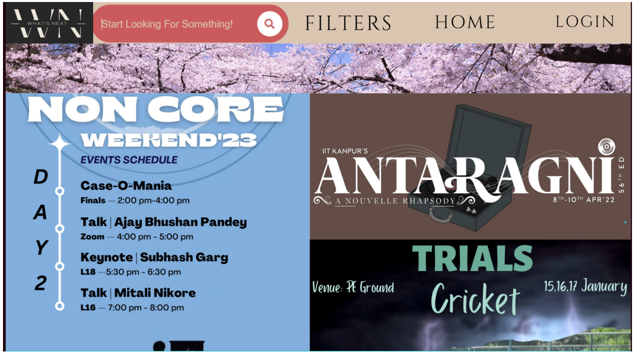
- At any point, there is a single account with admin privileges for regulating and deciding the special events.
- Our system is adopted by the LHC staff, who act as admins and can]]]update the schedule of the lecture halls on our servers.
- Upcoming technology will not affect the working of our product.
- The clients can only add events up to one month into the future.
- The no. of events that will be scheduled by various clients shall never exceed 1000.

3 Specific Requirements

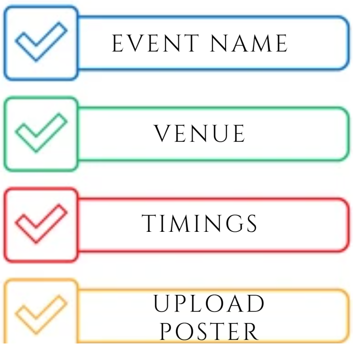
3.1 External Interface Requirements

3.1.1 User Interfaces

There will be two kinds of user interfaces - one for the admin and one for the general audience.
Landing page will have two tabs for running events - one each for academics and extra-curricular events. Also one tab will be there for overall timetable.

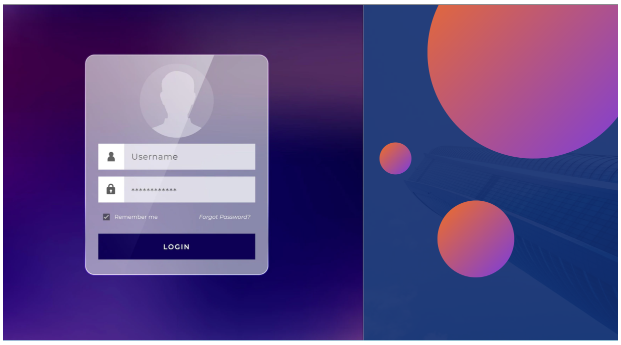


General Dashboard will be publicly accessible. It will contain the list of current and upcoming events divided according to categories.



admin Dashboard will be restricted only to authorised clubs. It will allow them to schedule their events either as an individual event or in a series. While scheduling an event, the system will inform about clashes, if any.
Admin Dashboard This will be accessible only to the website admin. It will feature super-user actions like approving a special event.
Super Admin Dashboard will be the interface visible to super admin to process requests put up by the admins such as scheduling of special events.

UI Design
Sign in-Sign up Page This portal is for the admins to login after which the dashboard will be visible to admins. **General Dashboard** diagram



3.1.2 Hardware Interfaces

Support for modem and Ethernet card - that is, appropriate drivers of compatible modem and Ethernet card are installed for accessing webpages over the internet or intranet.
The server side of the system must be connected to the IITK intranet.

3.1.3 Software Interfaces

The software shall run on almost all operating systems and their different versions. Javascript libraries like ReactJS and Express will be useful in making the web application.
We will be using the functionalities of third-party software such as Google Maps and Google Calendars. We are using MongoDB for databasing and We will be using two databases. One for events and one for creating admin login data fields.

3.2 Functional Requirements

3.2.1 The system shall provide an easy interface to the admins for scheduling upcoming events.

The admins should be able to schedule upcoming events by uploading the necessary details of them. The admin shall also be able to delete an event prior to its scheduled time. The admins shall also be able to provide the location of the event along with any relevant posters or brochure,etc. Also, the event can be scheduled as an individual event or in a series of events. The admin should be able to update the information about the event before the scheduled time. This information shall be visible to the audience on the event page.

3.2.2 The system shall help the admin in detecting any clash.

If the event time clashes with that of some other pre-scheduled event, the system shall notify about the clash to the admin. But it is the admin's discretion to schedule the event with clash or not.

3.2.3 The system shall allow the admin to schedule their event as special event.

There will be a provision of categorising a particular event as special, but for that the organiser will have to send request to the admin and only after the approval of request from the admin's side, the organiser will be able to tag that event as special.

3.2.4 The user shall be able to view the current and upcoming events sorted according to their scheduled time.

The user will be able to view the details of events that are planned by the admins along with the location, timing and map.

3.2.5 The system shall enable the users to view events filtered according to categories.

The events will be categorized under different verticals like programming, sports, cultural and special events and displayed under different tabs.

3.2.6 The system shall display an integrated schedule.

An integrated timetable (academics and co-curricular) of 7 days will be displayed under the integrated timetable section.

3.3 Use Case Model

3.3.1 Use Case #1 (U1)

Purpose- To explain how an admin can put up new events, and how those would be accessed by a general user.

Requirements Traceability- The admin must have logged in using his credentials.

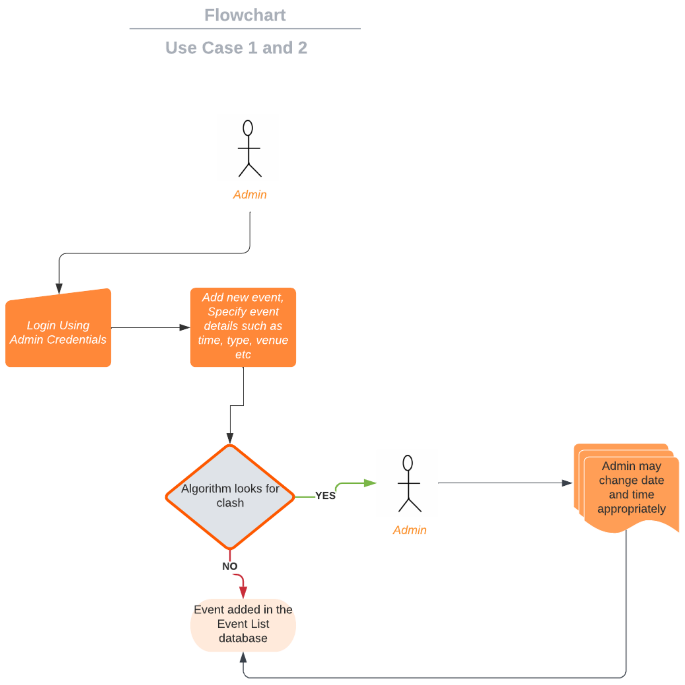
Priority- The priority of this use case is high, as adding and viewing new events is an integral part of the software.

Preconditions- The admin must have gone through the authentication process(admin login)

Post conditions- A new event will be added in the schedule. It would show up in the list of events particular to the admin who added it and would be visible to the general user depending on the timeline.

Actors- The actors involved in the use case are - admin for a club who adds events, and a general user who can see them.

Exceptions- Some clash warning may be given if there is another event scheduled at that time slot(or maybe even at same location).


Flowchart
Use Case 1 and 2
Admin
Login Using Admin Credentials
Add new event, Specify event details such as time, type, venue etc
Algorithm looks for clash
YES
Admin
Admin may change date and time appropriately
NO
Event added in the Event List database

3.3.2 Use Case #2 (U2)

Purpose- To explain the procedure that takes place when an admin adds an event but there is a clash of events involved.

Requirements Traceability- The admin must have logged in using his credentials.

Priority- The priority of this use case is medium, as it is upto the admin whether to accept the warning or not. He will still be able to add events in case the warning system fails.

Preconditions- In the time slot that the admin wants to schedule his/her event, at least one other event should be scheduled at the same time.

Post conditions- It is upto the admin to decide whether he wants to schedule his/her event in the same time slot or wants to change it.

Actors- The actor involved in the use case is the admin of a club who is adding an event in a slot that already has some other event scheduled.

Exceptions- Time slots may be such that clash duration is very small(for eg, only 10 minutes). In such a case, the warning will include the clash duration, and may even be skipped altogether.(doubt)

3.3.3 Use Case #3 (U3)

Purpose- To explain the process of a general user seeing the events of their interest

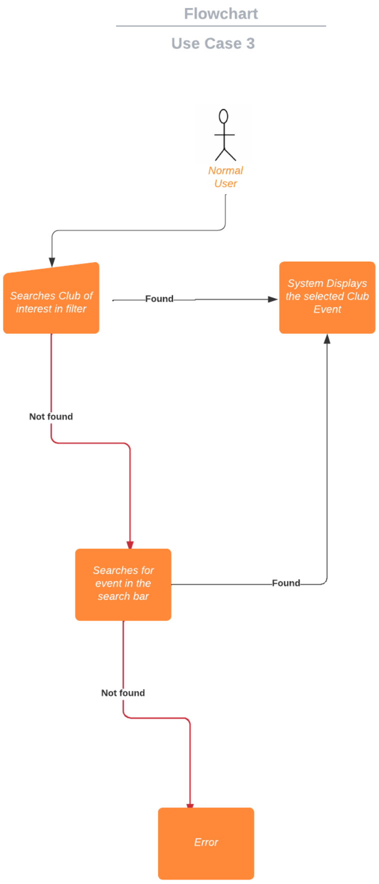
Priority- The priority of this use case is medium, as the user can anyways see the current and upcoming overall events.

Preconditions- There should be some events scheduled that match the interest of the user.

Post conditions- The user will be able to see the events particular to his interests(which he has to select).

Actors- The actor involved in the use case is the user who wants to see events of particular categories of his interest.

Exceptions- No events are scheduled which match the interest of our user, in which case he will just see a message stating that no such events are scheduled.


Flowchart
Use Case 3
Normal User
Searches Club of interest in filter
Found
System Displays the selected Club Event
Not found
Searches for event in the search bar
Found
Not found
Error

3.3.4 Use Case #4 (U4)

Purpose- To explain how an admin can schedule his event as a special event.

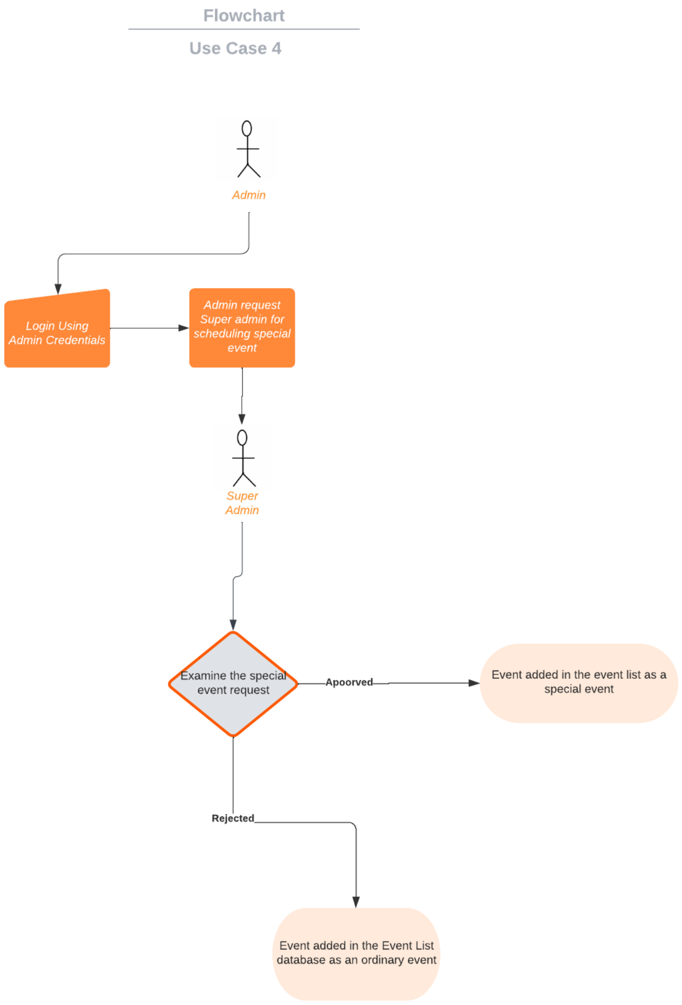
Requirements Traceability- The admin must have logged in using his credentials and the super-admin should be able to see requests for special events.(doubt)

Priority- The priority of this use case is medium, as the admin can schedule his event as a normal event too.

Preconditions- The admin must have a good reason for a special event.
(doubt)

Post conditions- Dependent on whether the super-admin decides to give permission for a special event. If permission is granted, the event will be displayed under the special events section

Actors- The actor involved in the use case is an admin and the super-admin.


Flowchart
Use Case 4
Admin
Login Using Admin Credentials
Admin request Super admin for scheduling special event
Super Admin
Examine the special event request
Apoorved
Event added in the event list as a special event
Rejected
Event added in the Event List database as an ordinary event

4 Non-functional Requirements

4.1 User Interfaces

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.

4.1.1 GUI

Labels and icons need to be in line with current trends. At the same time, the appearance of the website must be clear for viewing and care must be taken to minimize information cluttering. The user should only have to focus on one information source at a time for achieving good readability. Design choices must clearly convey the purpose of each element of the software and there should be no ambiguity in the function of any element for users.

4.1.2 API

The following APIs will be used :

- **Google Fonts:** For using font libraries
- **Google Maps:** To display the location of events, if specified by the admins
- **Google Calendar:** To display the activities scheduled for a particular day or week in an exportable format for sharing.

4.2 Performance Requirements

- **Speed:** The application must provide fresh data to the requesting client at a rate of no less than 1Hz.
- **Memory Requirements:** The software should use available modern optimization techniques to minimize memory and storage requirements.

4.3 Password Complexity

The user will not be allowed to set **weak passwords**. Weak passwords have been defined as passwords possessing any one of the following qualities:-

- Length less than 8 characters
- Containing solely lowercase/uppercase alphabets
- Having a commonly used phrase, examples of which will be stored in the database

4.4 Safety

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

4.5 Security

- There will be CAPTCHA system to avoid DDOS attacks.
- Secure storage of passwords will be done after hashing of the passwords. The system admin can only verify if the password is correct but won't know the password submitted by the user.

4.6 Reliability

The software must be designed to handle user flow on a level that will be sufficient to allow reliable access to the software when required. On average, the site should be reasonable be expected to deal with 20,000 to 50,000 requests.

4.7 Maintainability

The software must be designed such that:

- The architecture design, implementation and documentation of the software be such that they make the system reduce the maintenance overhead as much as possible
- The design of the software must allow efficient resolution of security defects, including updating the documentation and testing, must take a person two days at maximum.
- Updating the software and adding minor new features must take a person at most one week from start to finish, including testing and planning time.

5 Appendices

Specifies other useful information for understanding the requirements. All SRS documents should include at least the following two appendices:

5.1 Appendix A

- **USER CLASS**

Description	All the users of the software. Account creation and login possible only through IITK email-id.
Attributes	– **email**: string – **password**: string – **userId** integer
Operations	– register – login – logout – change password – delete account
Relationship	Each object of this class corresponds to a unique user. There are three types of users: Super Admin, Admins and Normal Users

- **SUPER ADMIN CLASS**

Description	Super Admin is above admin and they have the authority to add new Admins and authorize special events.
Class variables	– **name**: string –
Operations	– addNewAdmin – deleteAdmin – editEvents
Relationship	They can add Admins and have authority to classify events as Special Event when requested by Admin.

- **ADMIN CLASS**

Description	Admins are the heads of clubs that can add events of the corresponding club
Class variables	– **clubName**: string – **clubId**: integer – **clubMembers**: list of strings – **events**: list of eventIds
Operations	– addNewMember – deleteMember – requestSpecialEvent
Relationship	They can add new events, ie, create new objects of ”Events” class. This class also store list of events corresponding to concerned club.

- **NORMAL USER CLASS**

Description	General users of the app
Class variables	– **name**: string – **interests**: list of clubId's – **bookmarks**: list of eventId's
Operations	– addInterests – addBookmark – editProfile
Relationship	They can see the events and filter them on the basis of their interests.

- **EVENT CLASS**

Description	Class of events.
Attribute	– **eventId**: integer – **eventTimeline**: pair of Time datatype – **associatedClub**: a clubId – **eventName**: string – **location**: string – **overallDescription**: string – **comments**: list of pairs of string and userId – **likeCount**: integer
Operations	– addNewEvent – removeEvent – editEvent – addComment – addLike – addDislike
Relationship	Admin users can add events.

5.2 Appendix B

Date	Timings	Duration	Minutes
11/01	10:00PM-12:00AM	2hr	● Idea suggestions by everyone. ● Two main ideas were presented. ● One idea was a payment system for making payments easier for campus junta. ● The second idea was developing an application for providing information about various events at IITK.
13/01	6:30PM - 9:30PM	3 hr	● Preliminary Discussion on infeasibility and usefulness of ideas ● It was commonly felt that a great number of issues were faced during conduction of events. ● A software solution for this problem was seen as a very pressing need ● Finalisation of the idea of What's next.
19/01	7:00PM - 12:00AM	5 hr	● Brainstorming on our final idea and discussion on possible use cases, features, data flows. ● Exploration of possible implementations, challenges, tech stacks ● It was decided to use JavaScript for building the backend for our web application
24/01	9:00PM - 11:00 PM	2hr	● The system will have three access levels. ● The levels will be user, admin, and super-admin each with their separate functionalities.
25/01	7:00PM-11:30PM	4.5 hr	● The functionalities for each access level were decided upon. ● The UI/UX interface was finalized for each level.