Software Requirements Specification

for

ClubZen IITK

Version 1.0

Prepared by

Group 12 Group Name: Tiny Coders

|  |  |  |
| --- | --- | --- |
| Akash Biswas | 200074 | [abiswas20@iitk.ac.in](mailto:abiswas20@iitk.ac.in) |
| Avinash Prased | 200231 | [avinashp20@iitk.ac.in](mailto:avinashp20@iitk.ac.in) |
| Chandekar Vidish Vijay | 200291 | [cvvijay20@iitk.ac.in](mailto:cvvijay20@iitk.ac.in) |
| Harshit Gupta | 200429 | [guptah20@iitk.ac.in](mailto:guptah20@iitk.ac.in) |
| Jaya Meena | 200472 | [jayameena20@iitk.ac.in](mailto:jayameena20@iitk.ac.in) |
| Nikhil Verma | 200637 | [nikhilv20@iitk.ac.in](mailto:nikhilv20@iitk.ac.in) |
| Priyal Agrawal | 200730 | [priyalag20@iitk.ac.in](mailto:priyalag20@iitk.ac.in) |
| Priyanka Meena | 200731 | [priyankam20@iitk.ac.in](mailto:priyankam20@iitk.ac.in) |
| Shreyasi Mandal | 200956 | [shreyansi20@iitk.ac.in](mailto:shreyansi20@iitk.ac.in) |
| Vartika | 201089 | [vartikag20@iitk.ac.in](mailto:vartikag20@iitk.ac.in) |

|  |  |
| --- | --- |
| Course: | CS253 |
| Mentor TA: | Aishwarya Gupta |
| Date: | 01.02.2022 |

Content

Contents

[Contents](#Contents) ii

[Revisions](#Revisions) ii

1 [Introduction](#_Introduction) 1

1.1 [Product Scope](#_Product_Scope) 1

1.2 [Intended Audience and Document Overview](#_Intended_Audience_and) 1

1.3 [Definitions, Acronyms and Abbreviations](#_Definitions,_Acronyms_and) 2

1.4 [Document Conventions](#_Document_Conventions) 3

1.5 [References and Acknowledgments](#_References_and_Acknowledgments) 3

2 [Overall Description](#_Overall_Description) 4

2.1 [Product Overview](#_Product_Overview) 4

2.2 [Product Functionality](#_Product_Functionality) 5

2.3 [Design and Implementation Constraints](#_Design_and_Implementation) 5

2.4 [Assumptions and Dependencies](#_Assumptions_and_Dependencies) 5

3 [Specific Requirements](#_Specific_Requirements) 6

3.1 [External Interface Requirements](#_External_Interface_Requirements) 6

3.2 [Functional Requirements](#_Functional_Requirements) 7

3.3 [Use Case Model](#_Use_Case_Model) 9

4 [Other Non-functional Requirements](#_Other_Non-functional_Requirements) 19

4.1 [Performance Requirements](#_Performance_Requirements) 19

4.2 [Safety and Security Requirements](#_Safety_and_Security) 19

4.3 [Software Quality Attributes](#_Software_Quality_Attributes) 19

5 [Other Requirements](#_Other_Requirements) 21

[Appendix A – Data Dictionary](#Data_Dictionary) 22

[Appendix B - Group Log](#Group_Log) 24

Revisions

| Version | Primary Author(s) | Description of Version | Date Completed |
| --- | --- | --- | --- |
| v1.0 | Group 12 - Tiny Coders | SRS 1.0 | 30/01/2022 |

# 

# Introduction

## Product Scope

**Description-**

ClubZen IITK is a platform to integrate all the three Councils, i.e. Games and Sports Council, Media and Cultural Council, Academics and Career Council (their clubs and societies), and the three major fests (Antaragni, Udghosh, Techkriti ) in one place. It connects all the aspects of the institute’s co-curricular activities including workshops, meetings, forums, and social interaction.

**Objectives and Benefits-**

* It will help in the smooth conduction of orientation.
* It will act as a platform for providing students with essential information and updates about the various events and activities.
* It will help the user in keeping a track of the upcoming events and plan accordingly.
* The users will be able to see everything about the three councils and fests in one single app without having to toggle between systems so that they deliver consistent and meaningful service to each user.
* It will be instrumental in identifying and resolving student issues.
* It will help in promoting discussion among the students with similar interests.
* The user will find it easier to interact with the PoR holders through this website.
* It will help the users in exploring their interests and improving their skills.

## Intended Audience and Document Overview

**Intended Audience -**

This document can prove to be useful to different categories of people and cater to a wide range of audiences at the same time including **the entire developers, document writers, stakeholders, project managers, testers, users, approvers**.

The intent and purpose of the app for each stakeholder may vary significantly.

**Document Overview -**

This is a working document and, as such, is subject to change.

In its initial form, it is incomplete by definition and will require continuous refinement. Requirements may be modified and additional requirements may be added as development progresses and the product description becomes more refined. This information will serve as a framework for the current definition and future evolution of the product.

## Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Antaragni | Antaragni is the Annual Cultural Festival of the Indian Institute of Technology Kanpur, usually held in October. |
| CC | Abbreviation used for Computer Centre, IIT Kanpur |
| CSS | Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. |
| Database | Collection of all the information monitored by this system. |
| HTML | The HyperText Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. |
| Java | Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. |
| JavaScript | JavaScript, often abbreviated JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. |
| MySQL | MySQL is an open-source relational database management system. |
| PoR | Abbreviation for Position of Responsibility |
| React | React is a free and open-source front-end JavaScript library for building user interfaces based on UI components. |
| SSO | Abbreviation for Single Sign On |
| Stakeholder | Any person with an interest in the project who is not a developer. |
| Techkriti | Techkriti is an annual four-day inter-collegiate technical and entrepreneurship festival at IIT Kanpur |
| Udghosh | Udghosh is the annual sports meet of IIT Kanpur. |
| Unix | Unix is a family of multitasking, multi-user computer operating systems |
| User | Reviewer or Author |

## 

## Document Conventions

**Formatting Conventions -**

* **Arial** font size 11 is used throughout the document for text.
* The document maintains a 1’’ margin and is single-spaced throughout..
* Section titles use font size 23 and subsection titles use font size 17
* Comments are italicized and important words are made bold.

## References and Acknowledgments

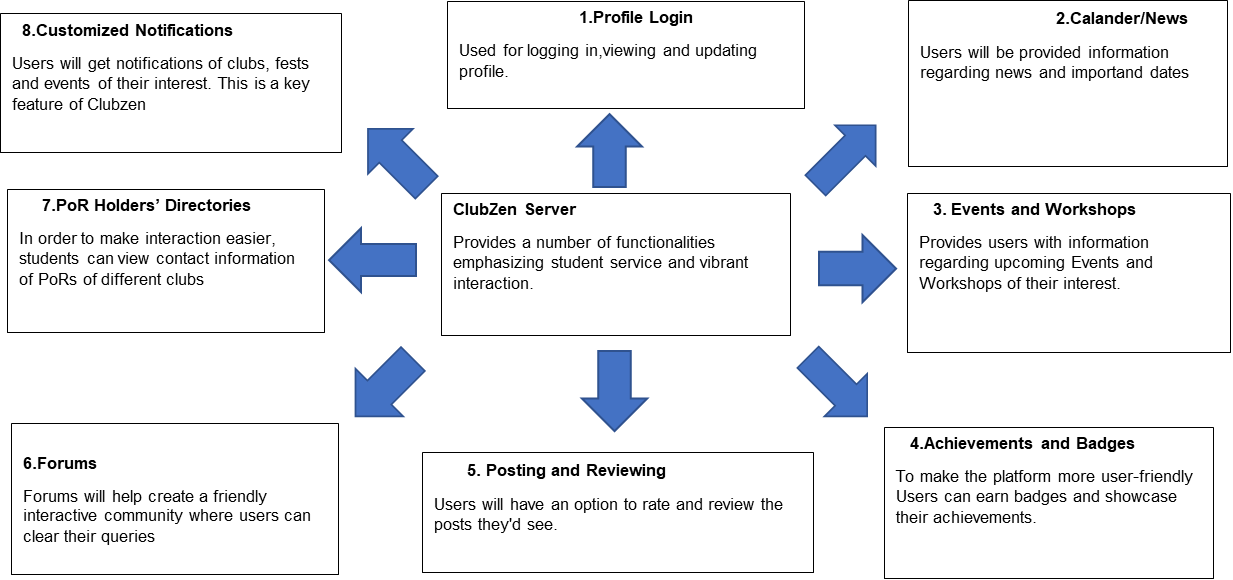
1. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.
2. Explore the UML sequence diagram - IBM

# Overall Description

## Product Overview

ClubZen is a website-based project which will be useful for college students and clubs and societies coordinators. At present, there is no official platform for notifications of clubs and societies. There are WhatsApp groups for different clubs and events, emails are sent to the IITK community but overall it is a complete mess. The students need to check all WhatsApp groups and emails for each and every activity of the clubs making it very difficult to find the required event-related details. To resolve this issue a website will be developed and this software will be for our college to get the events and workshops,  achievements, badges related information. Students can also contact the club’s secretaries and coordinators. They can post their works and get reviews from other students. We will also provide a forum for interaction among students.

In this project, the students and staff for different clubs are registered by the admin. The staff can create events for clubs and also post a timetable for the same. In this way, the students can see different events under sections for different clubs under the head of the website. This will be more helpful for the students as well as clubs also.



## Product Functionality

This product is intended to come with functionalities that emphasize student service and vibrant interactions. It includes the following features:

* Profile (Log in)
* Calendar/News
* Events and workshops
* Achievements and Badges
* Posting and Reviewing
* Forums
* PoR Holders’ Directories
* Customized Notifications

## Design and Implementation Constraints

The design & implementation constraints for this software product may include the following -

* The information of all users, clubs, and fests must be stored in a database that is accessible by the website.
* Users must have their correct usernames and passwords to log in to their accounts and do necessary operations.
* Response time for loading the product should take no longer than 2 minutes.
* Users may access from any computer that has Internet browsing capabilities and an Internet connection. General knowledge of basic computer skills is required to use the product.

## Assumptions and Dependencies

No specific assumptions or dependencies are considered at this time.

# Specific Requirements

## External Interface Requirements

### User Interfaces

The user interface shall be web-based, allowing users to remotely access the system via several applications. Users will be able to use the software through applications such as Microsoft Internet Explorer, Mozilla, Safari, Chrome et al.

Each part of the user interface intends to be as user-friendly as possible. The fonts and buttons used will be intended to be very fast and easy to load on web pages. The pages will be kept light in space so that it won’t take a long time for the page to load (React JS is used to ensure easy loading of the web page and prevent page refresh on navigation)

* Front-end software - HTML, CSS, JAVASCRIPT, REACT.JS
* Back-end software - JAVA

### Hardware Interfaces

* Hardware required to connect to the internet. For example, Modem, Network Card, Network Connection, etc.
* A device with an active internet connection. It can be a phone, a tablet, a desktop, anything with internet access, and a browser.

### Software Interfaces

During our system development, we have to design both static and dynamic website interfaces, create website functions and a database system, edit photos and pictures, and others, so it has a set of software requirements.

The following are needed requirements -

**Web browser (**with internet access**):** Used in the development phase for debugging and testing.

**Linux (Unix):** Operating system for a development environment.

**Java:** The language used to develop the backend for the progressive web application(**PWA**).

**HTML,** **CSS, JS:** A basic building block for the responsive web component - frontend of the web application

**React:** Helps the frontend interface become visually appealing for the users.

**MySQL:** Used to communicate to the server database.

**Visual Studio Code**: IDE to perform certain operations - writing code

## Functional Requirements

**3.2.1   Login with SSO**

The login screen will show two options - login with cc id or continue as a Guest.

Guests can only view the following - ( News, Events  with privacy set to “Public”)

If the User is the manager/Coordinator for any “Entity” (say TF) he/she gets the option to “Continue as TF” or “Continue as User”. A different interface is shown for entities. Entities can be Public/Private. Private entities need to approve every follower.

User Profiles through SSO -

Each user will have his/her profile with a profile picture and a few details like name and branch. It is possible to follow different clubs, societies to get updates about events they’re going to.

**3.2.2   Calendar / News**

The Calendar for the major bodies (Cult/Tech/any other body with a public calendar) will be available under this heading.

A personal calendar will also be available in which events that are subscribed/followed by the user will be displayed along with the option to add other non-app events.

Alongside with is a news section is displayed to inculcate any special news related to different clubs that will be available.

**3.2.3   Events and Workshops**

Events and workshops can be of three types-

**All** - All CC authenticated users

**Private** - Only those who follow the Entity (this makes sense if the Entity itself is private)

**Public** - All CC authenticated users as well as Guests

It will also be possible to restrict events to certain students within the institute (freshies/Sophie’s only or Elec/mech only or UG/PG only) and add multiple filters (such as Third Year Ph.D. Mech PGs only).

Entity Managers/Coordinators can create an event/Workshop as an Entity.

**3.2.4   Feedback**

There are also features to get feedback regarding the event. Initially, just a rating (1-5) is visible with the option to give a detailed review in paragraph form anonymously or non-anonymously. To comment/give feedback, the user must have a tag/ID.

**3.2.5   Achievements and Badges**

**Achievements:-**

In different sections of clubs, there is another sub-feature showing achievements and badges of the club/society/fest. Their different contributions towards various All India level events and their accomplishments. Members of the respective clubs/societies/fests can use these to customize their Official Page.

**Badges:-**

To make the platform more user-friendly there would also be a sub-feature that would award Users different badges depending on how interactive they are as well as which clubs, societies, and fests they interact with. User actions like rating, reviewing, liking and commenting would all contribute towards improving their positive score based on which they’ll eventually be awarded their badges. Users can use these badges to decorate their Profiles.

**3.2.6   Posting and Reviewing**

Here, there will be different sections for every club where different posts like their artworks, performances, etc can be posted after proper verification from different clubs entities.

A feedback review section will be there to like, comment, and review the posts.

**3.2.7  Forums**

A separate section where a user can ask or clear doubts regarding that particular club/society and members like POR holders of that club can clear the queries.

The aim is to break the ice between seniors and juniors and to celebrate healthy discussions.

**3.2.8  PoR Holders’ Directories**

The name, profile pic, and contact details for every PoR holder within the institute are arranged and shown in a structured manner with a subsection for each council.

**3.2.9 Customized notifications**

Every user will get customized notifications based on his/her interest i.e he/she will be notified based on the clubs followed by them regarding the activities ,posts etc.

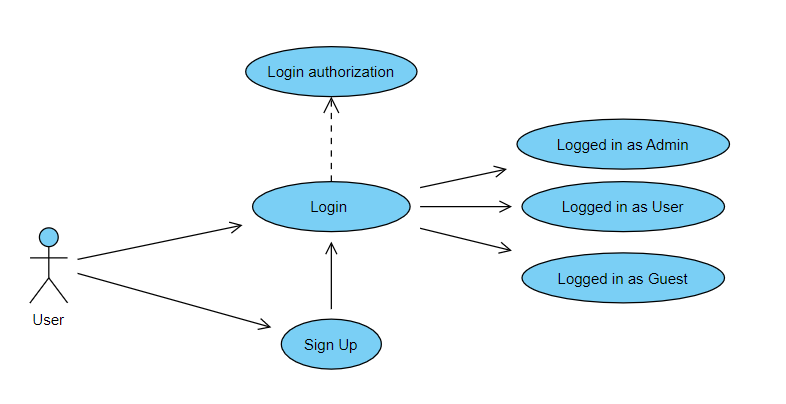
## Use Case Model

### 3.3.1     Use Case #1: Sign Up and Login

**Author –** Tiny Coders (team 12)

**Purpose** - This use case is for the login and signup of the user/admin/guest.

**Diagram**



**Requirements Traceability –** Profile page

**Priority** - High

**Preconditions** -

**Post conditions** - User is logged in

**Actors** – User of the website including the Manager/Coordinator.

**Exceptions** - User may forgot the password and go for forgot password option

**Includes** (other use case IDs)

**Notes -** Here the user will have to enter some necessary details (name, email etc.) during signup.

**3.3.2**     **Use Case #2: Forgot Password**

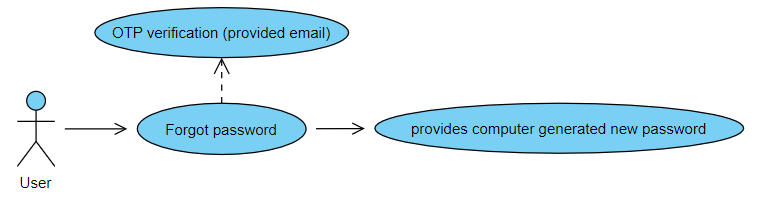
**Author –** Tiny Coders (team 12)

**Purpose** - This use case is for the updating of the user profile (name, profile picture and other details).

**Requirements Traceability – Profile** page

**Priority** – Low

**Diagram:**



**Preconditions** - User must not be logged in

**Post conditions** - User gets a computer-generated password

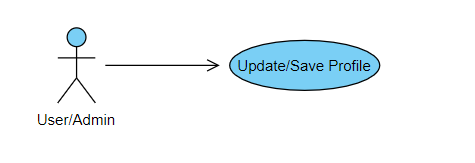
**Actors** – User of the website including the Manager/Coordinator.

**Exceptions** -

**Includes** (other use case IDs):

### 3.3.3     Use Case #3: Update Profile

**Diagram:**



**Author –** Tiny Coders (team 12)

**Purpose** - This use case is for the updating of the user profile (name, profile picture and other details).

**Requirements Traceability – Profile** page

**Priority** - Low

**Preconditions** - User must be logged on the website

**Post conditions** - User profile changes will be saved and reflected on the profile page

**Actors** – User of the website including the Manager/Coordinator.

**Exceptions** -

**Includes** (other use case IDs):

**3.3.4**     **Use Case #4: Creation of Club/Fest page**

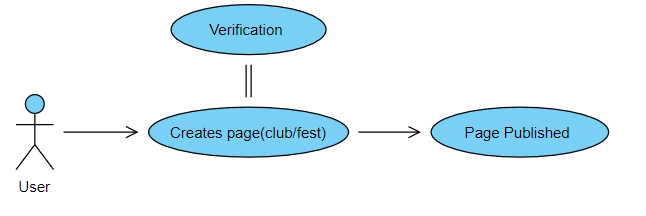
**Author –** Tiny Coders (team 12)

**Purpose** - This use case is for creating club/events pages.

**Requirements Traceability – Landing** page

**Priority** - High

**Diagram:**



**Preconditions** - User must be logged in the website and User must be the Coordinator/Manager of the club/fest.

**Post conditions** - The page for the respective club/fest is created and available for all the users.

**Actors** – User

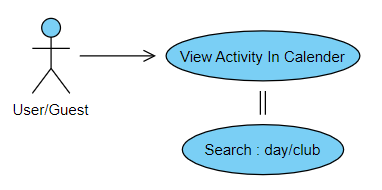
**Exceptions** - The page will not be created if the verification process fails

**Includes** (other use case IDs):

**Notes -** Any user can create a page for a particular club which then will be sent for verification from the respective body of ClubzenIITK.

### 3.3.5     Use Case #5: Viewing Activities in Calendar

**Diagram:**



**Author –** Tiny Coders (team 12)

**Purpose** - This use case is used for viewing the schedule of different activities (events and workshops) in the calendar.

**Requirements Traceability –** Calendar page

**Priority** - High

**Preconditions** - User must be logged on the website

**Post conditions** -User will get full data all the present and upcoming activities and events

**Actors** – User of the website including the Manager/Coordinator.

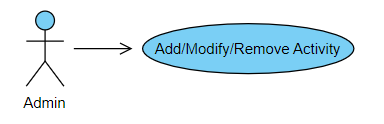
**Exceptions** -

**Includes** (other use case IDs):

**Notes -** We will provide a search bar next to the calendar so that the user can search for activities related to a particular club/fest or activities scheduled for a particular day.

### 3.3.6     Use Case #6: Updating Activities in Calendar

**Diagram:**



**Author –** Tiny Coders (team 12)

**Purpose** - This use case is for the updating Activity Calendar.

**Requirements Traceability – Calendar** page

**Priority** - High

**Preconditions** - User must be logged in the website as an admin.

**Post conditions** - The respective activities will be added/modified/removed from the calendar

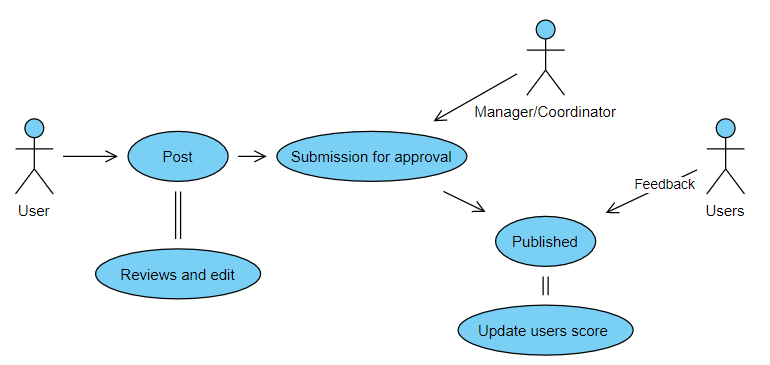
**Actors** – Admin

**Exceptions** -

**Includes** (other use case IDs):

**3.3.7**     **Use Case #7: Posting and Reviewing**

**Diagram:**



**Author –** Tiny Coders (team 12)

**Purpose** - This use case is triggered when the users want to post something to the club page to showcase their skills and get reviews from admins and other users.

**Requirements Traceability –** Landing page, Create post page

**Priority** - High

**Preconditions** - User must be logged in the website

**Post conditions** -User’s post will be published and all the users will be able to give feedback.

**Actors** – User of the website including the Manager/Coordinator.

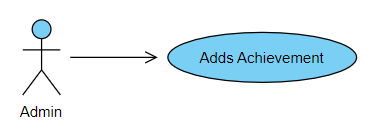
**Exceptions** -

**Includes** (other use case IDs):

**Notes -** Based on the users’ comments and posts we will update that user score for badges.

**3.3.8**     **Use Case #8: Add Achievement**

**Diagram:**



**Author –** Tiny Coders (team 12)

**Purpose** - This use case is for updating the Achievement and Badges section of the club/fest.

**Requirements Traceability –** Achievement/Badge section

**Priority** - Medium

**Preconditions** - Users must be logged on the website as an admin.

**Post conditions** - Achievement will be added

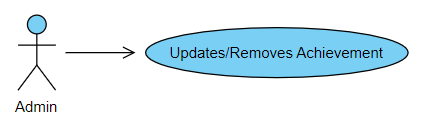
**Actors** – Admin.

**Exceptions** -

**Includes** (other use case IDs):

**3.3.9**     **Use Case #9: Update Achievement**

**Diagram:**



**Author –** Tiny Coders (team 12)

**Purpose** - This use case is for updating the Achievement and Badges section of the club/fest.

**Requirements Traceability – Achievement**/Badge section

**Priority** - Medium

**Preconditions** -User must be logged on the website as an admin.

**Post conditions** -Achievement will be updated

**Actors** – User of the website including the Manager/Coordinator.

**Exceptions** -

**Includes** (other use case IDs):

**3.3.10**     **Use Case #10: Forums**

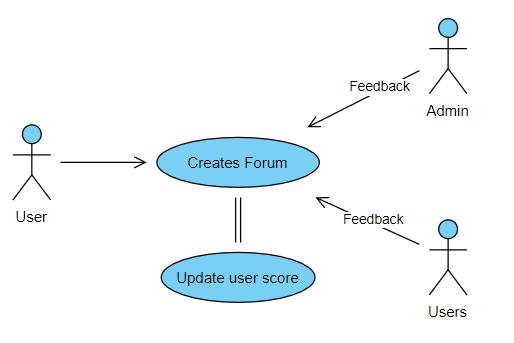
**Author –** Tiny Coders (team 12)

**Purpose** - This use case is triggered when the user wants to create a forum to interact with other users and admins.

**Requirements Traceability – Forum** section

**Priority** - High

**Diagram:**



**Preconditions** - User must be logged on the website

**Post conditions** -

**Actors** – User of the website including the Manager/Coordinator.

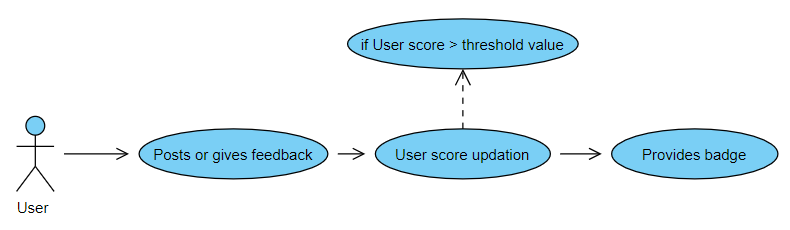
**Exceptions** -

**Includes** (other use case IDs):

**Notes -** Based on the users’ replies on the forum we will update that user score for badges.

**3.3.11**     **Use Case #11: Distribution of badges**

**Diagram:**



**Author –** Tiny Coders (team 12)

**Purpose** - This use case is triggered when the user posts or comments something on someone’s post or reply to forums.

**Requirements Traceability –**

**Priority** - medium

**Preconditions** -

**Post conditions** -

**Actors** – User of the website.

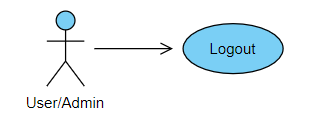
**Exceptions** -

**Includes** (other use case IDs):

**Notes -** When this use case is triggered it updates the user score and then checks if the user score is greater than the threshold value needed for a particular badge.

**3.3.12**     **Use Case #12: Logout**

**Diagram:**



**Author –** Tiny Coders (team 12)

**Purpose** - This use case is triggered when the user wants to logout.

**Requirements Traceability – Logout** link

**Priority** - High

**Preconditions** - User must be logged on the website

**Post conditions** -

**Actors** – User of the website including the Manager/Coordinator.

**Exceptions** -

**Includes** (other use case IDs):

# Other Non-functional Requirements

## Performance Requirements

The Product should provide maximum quick response to the requests of users while simultaneously handling multiple requests. Fast up-gradation of data should be available and the changes must be listened to and updated in real-time on the client's side. The system should be stable and maintainable, i.e. without any bugs and performance liabilities to function continuously. The underlying hardware and software should be able to handle more clients than the intended average number as the product platform can have a very wide range of users.

The product should be accessible from almost all platforms and should dynamically adapt itself according to the client’s specific hardware and software availability. Moreover, it should be responsive enough even on low spec hardware. This product should be dynamic. Changes and updates should be completed without a big shutdown of the system. Also, its scale should be dynamic too to cater to the ever-growing client community it will be catering to.

## Safety and Security Requirements

1. Passwords will be saved encrypted in the database to ensure the user's privacy.

2. The user's IP address will be logged.

3. The system will be protected against vulnerabilities such as SQL injection attacks.

4. Log in for IIT Kanpur students will be based on CC user ID and certain features will be restricted to general users.

## Software Quality Attributes

**4.3.1   Flexibility**

The application is to be designed so that it is flexible. Also, it should allow incorporating new requirements in any module of the system. The application will be designed in a modular format such that any future changes (additions or deletions) will be easily incorporated into the system.

**4.3.2   Portability**

The application will be easily portable on any window-based system. The website frontend is designed using React JS, making it a responsive and progressive web app, which ensures that the application can run on different platforms.

**4.3.3   Maintainability**

The architecture, design, implementation, and documentation of the product should minimize the maintenance costs of the software system. The maximum person-time required to fix a security defect (including regression testing and documentation update) must not exceed two persons per day.

Otherwise, the software system must be taken offline or the offending feature disabled which should only be done in case of emergencies. The average person-time required to make a minor enhancement (including testing and documentation update) must not exceed one person per week.

**4.3.4   Usability**

The frontend of the software application is designed to be user-friendly such that the user can utilize the system effectively. It will be based on the well-known principle of usability - Keep it Simple and Stupid.

**4.3.5   Security**

The application is password protected and also any upgradation of new entries and deletions is done by only privileged users. All the users will be able to log on to their accounts using their CC username and custom password (created by them). This password will be saved in an encrypted format in the database to ensure security.

In case the user forgets their password, security questions may be asked (set during the creation of an account), following which an email will be sent to his/her IITK email ID or other registered email address.

**4.3.6   Reliability and Availability**

The system shall provide a backup facility for databases on a regular basis to maintain data of users(frequenters / new) which may include passwords, recently uploaded posts, achievements, and activities of users.

The system shall also provide that it will be running  and ready to carry out its task whenever the user needs it.

# Other Requirements

Appendix A – Data Dictionary

**A.1 User Class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Element Name** | **Description** | **Operations** | **Requirement** |
| Email Address | It takes an Email address as an argument and associates it with an unique userId | Connect user ID to user | IIT Kanpur mail address |
| Password | It takes the password as arguments then checks the condition of login and returns a unique ID | Secure user login | string |
| User ID | System defines a unique ID for each user | Identifies unique user | unique string |
| User Profile Data | Maintain the data of each user | Store user info | string, int |
| Forum messages | User making forum to ask their doubts | Hold User Forum Message | string |
| Posts | Takes texts or image and post after approval by admin | User’s posts | abstract |

**A.1 User Interface Class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Element Name** | **Description** | **Operations** | **Requirements** |
| Registration | It takes the name, surname, username, password, email as arguments then checks the condition of registration and returns the status of registration. | First time registration and data input | string, int |
| Login/Logout | It takes username and password as arguments then checks the condition of login and returns the status of login. | Enables login/logout functionality for user | userId, username, password |
| Make comments | It takes comments and quotes as arguments and returns the status of making comments. | Enables comments functionality on posts | string |
| Like | It takes a quote as an argument and returns the status of liking the quote. | Enables user to like others posts | userId, user input |
| Follow | It takes the userId as an argument and returns the status of following the user. | Enables user subscription functionality | userId |
| Post | It takes the userId and post as argument and returns the status of post | Enables user to post | userId |
| Review | It takes the userId and post as an argument and allows the user to review it. | Enables user to review other posts | userId, post data |
| Write Forum | It takes the userId and the doubt of the user as an argument and returns the status of the forum. | Enables forum interaction | userId, page details |

**A.1 User Admin Interface Class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Element Name** | **Description** | **Operations** | **Requirements** |
| Profile information | It takes the userId and its data as arguments and returns profile information for public users. | Provides Admin contact info | userId, user data |
| Publish | It takes userId and Publish data as arguments and returns ready-made publish material. | Enables publish functionality  (restricted) | userId, publish data |
| Add Calendar Item | It takes the userId of admin and event data and returns the calendar item. | Enables modification of calendar | userId, calendar data, new data to push |
| Manage Page | It takes userId as argument and enables edit, modify and delete functionality for the whole page | Enables admin to control pages and administer them | page data, new page data |

Appendix B - Group Log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL. no.** | **Date** | **Timings** | **Venue** | **Description** |
| 1 | 11/01/2022 | 10 pm - 11:30 pm | Zoom | Brain-stormed various possible prospective ideas for the project. |
| 2 | 15/01/2022 | 10 pm - 11:30 pm | Zoom | Finalized the idea for the project and discussed various aspects of it. |
| 3 | 23/01/2022 | 5:30 pm - 6:30 pm | Zoom | Studied the SRS template given and distributed the work amongst the team members. |
| 4 | 24/01/2022 | 12:15 pm - 12:40 pm | Meet | First meet with the Teaching Assistant Ms. Aishwarya Gupta. Discussed some doubts regarding the SRS documentation. |
| 5 | 24/01/2022 | 12:40 pm - 1:20 pm | Meet | Redistributed the pending work after having clarified the doubts with the TA. |
| 6 | 24/01/2022 | 9 pm - 12 am | RM building | Explored some more functionalities for the product and progressed with the SRS document. |
| 7 | 25/01/2022 | 5 pm - 8 pm | RM building | Explored various implementation tools and software for the project. |
| 8 | 27/01/2022 | 5 pm - 8 pm | RM building | Progressed with SRS documentation work. |
| 9 | 28/01/2022 | 5 pm - 8 pm | RM building | Progressed with SRS documentation work. |
| 10 | 30/01/2022 | 5: 15 pm - 6 pm | Meet | Second Meet with the TA to discuss the drafted SRS document. |