# Remote Collaboration Management System

**PROJECT COLLABCALL**



DELHI TECHNOLOGICAL UNIVERSITY

Project Report submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology.

in

**Computer Science & Engineering**

By

# MANISH SINGH , 23/CS/244

**Certificate**

This is to certify that project report entitled Collaboration Management System, submitted by Piyush Virmani in partial fulfillment for the award of degree of Bachelor of Technology in Computer Science & Engineering to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

**Date: Ms. Reema Aswani**

**Assistant Professor**

## Acknowledgement

There are many people who are associated with this project directly or indirectly whose help and timely suggestions are highly appreciable for completion of this project. First of all, I would like to thank Prof. Dr. SP Ghrera, Head, Department of Computer Science Engineering for his kind support and constant encouragements, valuable discussions which is highly commendable.

I would like to express my sincere gratitude to my supervisor Ms. Reema Aswani, for her super vision, encouragement, and support which has been instrumental for the success of this project. It was an invaluable experience for me to be one of her students. Because of her, I have gained a careful research attitude.

Lastly, I would also like to thank my parents for their love and affection and especially their courage which inspired me and made me to believe in myself.

Date: **Piyush Virmani**

**111213**

**CONTENTS**

**S. No. Topic Page No.**

* 1. **What is a Collaboration 1**

* 1. **INTRODUCTION**  **3**

2.1 System Perspective 4

2.2 System Features 5

2.3 User Characteristics 5

2.4 Operating Environment 5

2.5 Design and Implementation Constraints6

**3. SYSTEM FEATURES 7**

3.1. Registration 7

3.2. Single Login Authentication 8

3.3. Collaboration Request 8

3.4. Periodical Announcements 9

3.5. Collaboration Rooms 11

3.6. Web Page Announcements 12

3.7. Multiple Collaboration 12

# 4. EXTERNAL INTERFACE 14

4.1. User Interfaces 14

4.2. Hardware Interfaces 14

4.3. Software Interfaces 14

4.4. Communications Interfaces 14

# 5. NON-FUNCTIONAL REQUIREMENTS 15

5.1. Performance Requirements 15

5.2. Safety Requirements 15

5.3. Security Requirements 16

5.4. Software Quality Attributes 16

5.5. GNU GPL License 16

# 6. SOFTWARE SYSTEM ATTRIBUTES 17

6.1. High Availability 17

6.2. Scalability 17

6.3. Reliability 17

6.4. Usability 18

6.5. Portability 18

6.6. Extensibility 19

6.7. Maintainability 19

6.8. Accessibility 19

6.9. Security 20

1. **DESIGN CONSTRAINTS 21**

* 1. Design Objectives 21
  2. Design Consideration 22

1. **PERFORMANCE REQUIREMENTS 24**

1. **PORTAL FUNCTION REQUIREMENTS 25**

# 10. APPENDIX 26

10.1. Glossary 26

10.2. System Requirements 27

10.3. Function Requirements 32

10.4. Server Requirements 35

10.5. Hosting Requirements 36

**11. COLLABORATION MANAGEMENT DIAGRAMS 38**

# List of Figures

**S.No. Title Page No.**

1. **Registration Page** **47**

1. **Login Page** **47**

1. **User Login Page 48**

1. **Collaboration Table 48**

1. **Collaboration Register Page 49**

1. **Feedback Form 49**

1. **Admin Page** **50**

1. **Add Collaboration Topic 50**

# Abstract

This project deals with the Collaboration management system .As a students or staff members are required to view the details of Collaboration is going to conduct in various colleges or institutions and to attend the Collaboration to gain knowledge from the Collaborations .Administrator will add the details about the various Collaborations available to attend for various department students and staff members. User will enter into the system by giving the username and password and selection form will be displayed for the user from that department should be selected and depending up on the department the Collaboration management system will show the details of the Collaborations in various place using My Sql and JSP. This web application will support every aspect of the Collaboration organization process. This includes paper submission, registration handling of the Collaboration participants, searching, downloading papers, management multiple Collaborations at a time .

System should be applicable to any scientific/technical Collaboration.

# CHAPTER 1

**What is a Collaboration ?**

A **Collaboration** is a like a conference of people who "confer" about a topic. Collaborations can be-

* **Academic Collaboration** in science and academic, a formal event where researchers present results, workshops, and other activities.
* **Business Collaboration** organized to discuss business-related matters
* **Collaboration call** in telecommunications, a "multi-party call"
* **Collaboration hall** room where Collaborations are held
* **News Collaboration** an announcement to the press (print, radio, television) with the expectation of questions, about the announced matter, following.
* **Parent-teacher Collaboration** a meeting with a child's teacher to discuss grades and school performance.
* **Peace Collaboration** a diplomatic meeting to end conflict.
* **Settlement Collaboration** a meeting between the plaintiff and the respondent in lawsuit, wherein they try to settle their dispute without proceeding to trial
* **Trade Collaboration or trade fair**, organized like a business Collaboration but with wider participation and providing the opportunity for business people and the general public alike to network and learn more about topics of interest through workshops, viewing whitepaper presentations, and meeting vendors of similar or related services.

Most Collaborations have one or more keynote speakers who will deliver the keynote speech. These are common at academic and business Collaborations. The speakers chosen are eminent personalities in the related field and their presence is meant to attract more people to attend the Collaboration. There are various types of Collaborations:

* A **symposium** is a casual gathering and includes refreshments and entertainment.
* A **seminar** is organized to discuss a particular topic. They are usually educational in nature and attendees are expected to gain new knowledge or skills at the end of the seminar
* A **workshop** is more of a hands-on experience for the participants with demonstrations and activities; the amount of time one speaker addresses the group is limited
* A **round-table** Collaboration is a get-together of peers to exchange thoughts and opinions on a certain topic, usually political or commercial. There are a limited number of participants who sit at a round table, so that each one can face all the others

For the smooth running of a Collaboration, meticulous planning must be carried out well in advance. All important aspects of the Collaboration must be covered, so it is better to maintain a checklist. A back-up plan to handle emergencies is always mandatory. The planning works better when individuals in the planning and administrative committee have clear roles and responsibilities assigned.

Firstly, the purpose of the Collaboration must be clearly understood. The budget needs to be defined. While some companies have sufficiently large Collaboration rooms to accommodate the event, an external venue is most commonly needed. The advantage of using a Collaboration room in a hotel is that accommodation costs might be offered at a discounted rate. One needs to negotiate for the lowest price at several venues that meet the specifications of the Collaboration. eVenues.com offers online search and booking facilities for all types of events, especially Collaborations.

Once the date, time and venue of the Collaboration are fixed, the availability of all presenters, vendors, attendees and special guests needs to be confirmed. Registration forms have to be sent out and as the responses come in, they must be recorded to get the confirmed number of attendees. This is needed to take care of the logistics for the Collaboration, such as flight bookings, accommodation and car pick-ups for important attendees like the keynote speaker and special guests. Based on the theme of the Collaboration, the invites, agendas and brochures have to be designed. Sponsors' names and logos must be included in these leaflets for advertizing.

**CHAPTER 2**

# INTRODUCTION

A **Collaboration management system** is web-based software that supports the organization of Collaborations especially scientific and Innovative Collaborations. It helps the program chair(s), the Collaboration organizers, the authors and the reviewers in their respective activities also helps in conference management also because it a plateform where anyone can create a thread workspace where they can create their agenda or problem statements their meeting guidelines and other important details and deadline and just by sharing the link they can access that workspace collab and contribute.

For faculty and researchers, attending at least one academic Collaboration annually in their fields of interest is inevitable. In such Collaborations, many stakeholders are involved in various Collaboration tasks. These include, but are not limited to, program committee chair, program committee members (reviewers), general chair, publicity chair, and authors . For a Collaboration organization to be successful, a process should be in place. The process of Collaboration organization consists of many phases, such as call for papers, paper submission, paper review, review discussion, paper re-submission, and author notification . Stakeholders with varying viewpoints, in addition to the complex Collaboration organization process, make organizers, especially those without any prior professional organization skills, feel unenthusiastic about managing an academic Collaboration, and possibly quit the task. With the presence of advanced technology affecting all perspectives of our life, academic Collaborations are increasing in great number. This is accompanied by an enormous increase in the number of submitted papers. To cope with such large a number of papers and to keep reviewing loads manageable, the number of program committee members has to significantly increase. Consequently, scheduling a face-to-face program committee meeting to review and confer paper submissions is deemed impractical. Based on what is mentioned above, it is vital to develop an online Collaboration management system that facilitates the task of Collaboration organization.

During the last several years Collaboration management systems (CMS) became Collaboration chairs’ best friend. As a web based information systems they offer a reliable user-friendly service anywhere at any time. Authors can easily track the status of their papers. Program Committee (PC) members can review them anywhere in the world. But the highest benefit is for PC chairs as the Collaboration management systems not just offer a user-friendly way of communication and data storage, but automation of a series of hard to handle and time consuming processes like assignment of reviewers to papers, conflict of interest detection, plagiarism detection and etc.

Collaboration management systems usually handle the entire process of Collaboration management in smaller time-dependent pieces called phases. Some phases can overlap in time while others should be strictly sequential as they use data submitted or generated within previous phase(s). There are activities related to Collaboration organizing (for example finding financial support; arranging halls for session presentations; arranging official dinners and other social activities; and etc.) that are more or less independent on the papers' management so they could be performed in parallel. At the time of assignment all papers have to be already submitted, all reviewers already registered and all bids stated so that the automatic assignment module proposes the best possible assignment taking into account both - selected topics and reviewers’ bids. The reviewing and discussion phases can slightly overlap as some papers may already have all reviews done and waiting for discrepancy resolution while other papers may still be waiting for evaluation. The process of assignment of reviewers to papers is probably the most important and challenging one. Its accuracy directly impacts the quality of the Collaboration and its image. For high level Collaborations, having a low acceptance ratio, it is crucial that papers are evaluated by the most competent in the relevant subject domain reviewers. The assignment could be performed both manually and automatically. Manual assignment is feasible for Collaborations having a small amount of submitted papers. However when the number of papers and reviewers increases the manual assignment gets less and less accurate due to the constraints it should satisfy – high accuracy, no conflict of interest and not on the last place – load balancing (i.e. all reviewers should evaluate roughly the same number of papers).

## 2.1 System Perspective

Collaboration management system manages all administrative and organizational tasks of a Collaboration. The system is a server side web application that uses external relational database management system as a data tier. Every user of this system needs only a web browser to use the system and connect to it. The system will be based on Nodejs and Express for backend and for frontend we uses React ,Vite , TailwindCss ,HTML and CSS technologies which provide easy and rapid development. This system will also be able to host multiple Collaborations and their all web based activities.

## 2.2 System Features of CollabCall

* **The brief Overview** :
* The CollabCall platform is designed with a variety of features that enhance user interaction, facilitate collaboration, and streamline project management. Below is a comprehensive overview of the system features:
* **1. User Management**
* **User Registration and Login:**
* Secure account creation and authentication via email or social media integration.
* **User Profiles:**
* Personal profiles that display user information, roles, and contributions to various projects.
* **2. Thread Workspace Creation**
* **Create New Workspace:**
* Users can easily create new thread workspaces by providing a title, problem statement, motives, deadlines, and relevant tags for better organization.
* **Templates:**
* Pre-defined templates to guide users in structuring their workspaces effectively.
* **3. Collaboration Tools**
* **Discussion Forum:**
* A threaded discussion area where users can post comments, ask questions, and collaborate in real-time.
* **Real-Time Editing:**
* Multiple users can edit shared documents or proposals simultaneously, with changes reflected instantly.
* **Voting System:**
* Users can vote on proposed ideas or solutions, helping to prioritize contributions and efforts.
* **Mentorship Requests:**
* Users can request advice or guidance from experts within the platform.
* **4. Project Management Features**
* **Task Assignment:**
* Users can create tasks within the workspace, assign them to specific team members, and set deadlines.
* **Progress Tracking:**
* A dashboard that allows users to see the status of tasks, deadlines, and overall project progress.
* **Milestone Setting:**
* Users can set important milestones to track significant phases of the project.
* **5. File Sharing and Resource Management**
* **Document Upload:**
* Users can upload and share documents, images, and other resources relevant to their projects.
* **Resource Hub:**
* A central repository of tools, articles, and links that can assist users in their problem-solving efforts.
* **6. Notifications and Alerts**
* **Real-Time Notifications:**
* Users receive alerts for new comments, task assignments, upcoming deadlines, and updates in their workspaces.
* **Email Notifications:**
* Options for users to receive email summaries or alerts regarding workspace activities.
* **7. Analytics and Reporting**
* **Project Analytics:**
* Tools to analyze the progress and participation rates within workspaces, helping users understand engagement levels.
* **Feedback Collection:**
* Users can provide feedback on the workspace experience, which can be analyzed for continuous improvement.
* **8. Integration and API Access**
* **Third-Party Integrations:**
* Ability to integrate with other tools and platforms (e.g., Google Drive, Slack, Trello) for enhanced functionality.
* **API Access:**
* An API for developers to create custom applications or integrations with the CollabCall platform.
* **9. Security and Data Privacy**
* **Data Encryption:**
* All user data and communications are encrypted to ensure privacy and security.
* **Role-Based Access Control:**
* Different access levels for users based on their roles within a project (e.g., admin, member, viewer).
* **10. User Support and Documentation**
* **Help Center:**
* A dedicated section with FAQs, guides, and tutorials on using the platform effectively.
* **Customer Support:**
* Access to support via chat, email, or ticketing system for resolving user issues.
* **Conclusion**
* The features of CollabCall are designed to create a comprehensive, user-friendly platform that encourages collaboration, enhances project management, and facilitates community-driven problem-solving. By incorporating these features, CollabCall aims to empower users to engage actively in addressing real-world challenges and promote effective teamwork.

## 2.3 User Characteristics

Understanding the user characteristics of CollabCall is essential for designing an effective platform. The target users are diverse and can be segmented into different categories based on their roles and needs:

**1. Target User Groups**

* **Students and Educators:**
  + Engaged in academic projects and looking for collaborative solutions to real-world problems.
  + May require guidance on project management and problem-solving techniques.
* **Professionals and Practitioners:**
  + Individuals working in various industries who seek to collaborate on projects, share knowledge, and solve specific problems.
  + Value efficient communication, task management, and the ability to share resources.
* **Community Organizers and Activists:**
  + Users focused on social issues and community engagement, looking for a platform to rally support and collaborate on initiatives.
  + Require tools for organizing events, tracking progress, and mobilizing resources.
* **Tech Enthusiasts and Developers:**
  + Individuals interested in leveraging technology to create solutions, who may also contribute technical expertise to projects.

**2. User Skills and Technical Proficiency**

* **Varied Technical Backgrounds:**
  + Users may range from tech-savvy individuals familiar with collaborative tools to those with limited technical experience.
  + The platform should cater to both ends of the spectrum by offering intuitive interfaces and guided workflows.
* **Problem-Solving Skills:**
  + Users should possess varying degrees of problem-solving skills; the platform can support skills development through resources and mentorship features.

**3. User Goals and Motivations**

* **Collaboration:**
  + Users seek to work together to tackle issues, share knowledge, and achieve common goals.
* **Learning and Development:**
  + Many users are motivated by the opportunity to learn from others, develop new skills, and gain insights into effective problem-solving.
* **Impact:**
  + Users are driven by the desire to make a positive impact in their communities or fields of interest.

## 2.4 Operating Environment

The operating environment of CollabCall encompasses the technical and contextual aspects that influence how the platform functions and is utilized by its users.

**1. Technical Environment**

* **Web-Based Platform:**
  + Accessible through web browsers on various devices (desktops, laptops, tablets, and smartphones).
  + Ensures that users can collaborate from anywhere with an internet connection.
* **Scalability:**
  + Built to handle a growing number of users and projects, ensuring performance and responsiveness as the user base expands.
* **Integration Capabilities:**
  + Ability to integrate with third-party applications (e.g., Google Drive, Slack) and APIs, enhancing functionality and user experience.

2.5 Design and Implementation Constraints for ***CollabCall***

As the creators of CollabCall, we have developed a collaborative platform that addresses various user needs and challenges in problem-solving and project management. Throughout the design and implementation process, we encountered several constraints that shaped our approach. Here’s an overview of these constraints along with the innovative reverse engineering process we employed.

**1. Design Constraints**

**User Interface (UI) Design**

CollabCall features an intuitive and user-friendly interface that prioritizes simplicity and usability. We designed the UI to be easy to navigate, ensuring that users, regardless of their technical expertise, can engage with the platform effectively. The responsive design allows seamless access across various devices, including desktops, tablets, and smartphones, enhancing user experience.

**Accessibility Standards**

To ensure inclusivity, CollabCall adheres to accessibility guidelines (such as WCAG 2.1). This commitment allows users with disabilities to fully interact with all functionalities of the platform, ensuring that everyone can benefit from our collaborative tools.

**Branding and Aesthetics**

The platform reflects a consistent brand identity, with a cohesive color scheme and typography that enhances visual appeal. This aesthetic consistency fosters a sense of professionalism and trust among users.

**2. Implementation Constraints**

**Technical Constraints**

CollabCall is built using a carefully chosen technology stack, including React for the frontend and Node.js for the backend. This selection aligns with our team's expertise and the project’s requirements, enabling us to deliver a robust and efficient platform. Additionally, we have integrated with various third-party services, ensuring compatibility while offering enhanced functionality.

**Performance Constraints**

We designed CollabCall with scalability in mind, accommodating a growing user base and numerous projects without compromising performance. Optimizing load times was a priority; therefore, we implemented strategies to minimize delays, ensuring users have a smooth experience.

**Security Constraints**

Data protection is paramount for CollabCall. We comply with data protection regulations such as GDPR, which guides how we collect, store, and process user data. Our platform features secure user authentication methods, such as OAuth, to safeguard user information while maintaining ease of access.

**Resource Constraints**

Given the limitations of budget and time, we prioritized core functionalities for the initial launch of CollabCall. This strategic approach allows us to focus on essential features while planning for future enhancements. Our team's expertise also guides the complexity of features we can implement effectively.

**3. Regulatory Constraints**

CollabCall adheres to relevant legal standards for software applications, including copyright laws and user data protection regulations. This compliance ensures that we respect intellectual property rights and user privacy.

**4. User Feedback and Iteration Constraints**

We recognize the importance of user-centric design and actively seek feedback to improve the platform. However, we carefully balance the implementation of changes based on feedback with our development timelines and available resources.

**CHAPTER 3**

# SYSTEM FEATURES

Detailed Features of CollabCall

CollabCall is designed as a comprehensive collaborative platform that facilitates community-driven problem-solving and project management. Below is a detailed description of each key feature:

**1. Thread Workspace Creation**

**Description:**

The thread workspace is a dedicated area where users can articulate and discuss specific problems or projects. Each workspace functions as a mini-project hub, allowing teams to organize their discussions and resources effectively.

**Key Components:**

* **Problem Statement Submission:** Users can submit a clear and concise problem statement, providing context and background to frame the discussion.
* **Motive or Agenda Description:** A section for users to elaborate on the significance of the problem, outlining their goals and what they hope to achieve through collaboration.
* **Important Dates and Deadlines:** Users can set deadlines and important milestones to keep the team on track and focused on goals.

**Benefits:**

This feature provides structure to discussions, ensuring that all team members understand the problem at hand and the objectives of their collaboration.

**2. Discussion Forum**

**Description:**

The discussion forum is a threaded area where users can engage in conversations related to the problem statement. This forum allows for organized discussions, enabling users to communicate effectively without losing track of important points.

**Key Components:**

* **Comment Threads:** Users can post comments, questions, and suggestions, creating a threaded conversation that is easy to follow.
* **Mentions:** Users can tag other participants to draw their attention to specific comments or questions.
* **Search Functionality:** Users can search through discussions to find relevant information or previous conversations.

**Benefits:**

This feature fosters active engagement and collaboration, making it easier for users to share insights and brainstorm solutions.

**3. Real-Time Editing**

**Description:**

Real-time editing allows multiple users to work on documents or proposals simultaneously. This feature is essential for collaborative efforts, ensuring that all team members can contribute to the project in real-time.

**Key Components:**

* **Simultaneous Editing:** Users can edit the same document at the same time, with changes reflected instantly for all participants.
* **Change Tracking:** Users can see who has made changes and when, providing transparency in collaboration.

**Benefits:**

This feature enhances teamwork by allowing users to collaborate seamlessly, reducing the time spent on revisions and document management.

**4. Task Assignment and Progress Tracking**

**Description:**

This feature allows users to create and assign specific tasks within the workspace, helping teams organize their efforts and monitor progress.

**Key Components:**

* **Task Creation:** Users can create tasks with descriptions, deadlines, and priority levels.
* **Assignment:** Tasks can be assigned to specific team members, ensuring accountability.
* **Progress Monitoring:** Users can update the status of tasks (e.g., “In Progress,” “Completed”), allowing the team to track overall project progress.

**Benefits:**

Task assignment and progress tracking keep the team organized and focused on their objectives, facilitating better project management.

**5. File Sharing and Resource Management**

**Description:**

This feature enables users to upload and share relevant documents, images, and resources within the workspace, creating a centralized repository for all project-related materials.

**Key Components:**

* **File Uploads:** Users can upload files directly to the workspace.
* **Link Sharing:** Users can share links to external resources or tools that are pertinent to the project.
* **Organized Folders:** Files can be organized into folders for easier navigation and retrieval.

**Benefits:**

Centralized file sharing enhances collaboration by ensuring that all team members have access to the necessary resources, reducing confusion and streamlining communication.

**6. Voting System**

**Description:**

The voting system allows users to propose ideas or solutions within the workspace and enables other users to vote on them. This feature helps prioritize contributions based on community interest and consensus.

**Key Components:**

* **Idea Submission:** Users can submit ideas or proposals for consideration.
* **Voting Mechanism:** Other users can vote on submitted ideas, indicating their preference or support.
* **Results Display:** The platform displays the results of the voting process, highlighting the most supported ideas.

**Benefits:**

This feature empowers users to contribute actively and ensures that the most valued ideas are prioritized, fostering a sense of ownership and collaboration in the decision-making process.

**7. Notifications and Alerts**

**Description:**

The notifications and alerts system keeps users informed about important updates, deadlines, and activities within their workspaces.

**Key Components:**

* **Real-Time Notifications:** Users receive alerts for new comments, task assignments, and upcoming deadlines.
* **Email Summaries:** Users can opt to receive email summaries of activity within their workspaces, ensuring they stay informed even when not logged in.

**Benefits:**

This feature enhances user engagement by ensuring that team members are aware of developments within the project, promoting timely responses and active participation.

**8. Feedback Mechanism**

**Description:**

The feedback mechanism allows users to provide ratings and comments on their experience with the thread workspace, helping identify areas for improvement.

**Key Components:**

* **Rating System:** Users can rate their experience on a scale (e.g., 1 to 5 stars).
* **Comment Section:** Users can leave detailed feedback, suggesting improvements or highlighting issues.

**Benefits:**

This feature helps the CollabCall team gather insights for continuous improvement, ensuring the platform evolves to better meet user needs.

**9. Analytics and Reporting**

**Description:**

Analytics and reporting tools provide insights into user activity, project progress, and engagement levels within the platform.

**Key Components:**

* **User Engagement Metrics:** Track how often users participate in discussions, submit ideas, and complete tasks.
* **Project Performance Reports:** Generate reports on project status, including completed tasks and milestones achieved.

**Benefits:**

This feature enables users and administrators to assess the effectiveness of collaboration efforts, identify trends, and make informed decisions for future projects.

**10. Integration with Third-Party Tools**

**Description:**

CollabCall allows integration with popular third-party tools and services, enhancing its functionality and user experience.

**Key Components:**

* **Integration Options:** Support for linking with applications such as Google Drive, Slack, and Trello.
* **API Access:** Developers can create custom integrations, expanding the platform’s capabilities.

**Benefits:**

This feature increases CollabCall's versatility, allowing users to leverage their existing tools and workflows, ultimately improving productivity and collaboration.

**11. Reverse Engineering Process**

In developing CollabCall, we employed a **reverse engineering process** that greatly influenced our design choices. This innovative approach included:

**Analysis of Similar Platforms**

We closely examined existing collaborative platforms like Slack, Trello, and Asana to gain insights into their features, design choices, and user interactions. For example:

* **Slack:** We noted the effectiveness of its threaded messaging system, which allows users to follow specific conversations without cluttering the main chat. This inspired our discussion forum feature, where users can comment on problem statements in a structured manner.
* **Trello:** The visual task management approach in Trello, using boards and cards, demonstrated how users appreciate clear visual organization. This influenced our decision to incorporate task assignment features that allow users to visualize project progress.

**Critical Evaluation**

Each feature we observed was critically evaluated to understand its purpose and effectiveness. We asked ourselves vital questions, such as:

* **Why was this feature implemented?**
* **What user needs does it address?**
* **How does it enhance collaboration or project management?**

For instance, the voting system in platforms like Product Hunt was analyzed. We found that allowing users to vote on ideas helps prioritize contributions and encourages engagement. This insight led us to implement a similar voting feature within CollabCall to empower users to identify the most valuable solutions.

**Informed Design Choices**

The insights gained from this analysis informed our design and functionality decisions for CollabCall. By adopting best practices while avoiding common pitfalls, we have created a platform that effectively meets user needs and encourages collaboration. For instance, we recognized the importance of real-time updates from our analysis of Google Docs, which allows multiple users to collaborate seamlessly. This inspired us to integrate real-time editing capabilities into our platform, enhancing user interaction and teamwork.

**Conclusion**

The features of CollabCall are designed to create a comprehensive, user-friendly platform that enhances collaboration, project management, and community engagement. By integrating these detailed features, CollabCall empowers users to work together effectively in addressing real-world challenges, promoting active participation and innovative problem-solving.

Rewrite

**CHAPTER 4**

# EXTERNAL INTERFACE REQUIREMENTS

## 4.1. User Interfaces

User interface will be provided through a web browser as a web site. Thus the user interface will be constrained by the web browser capabilities.

**4.2. Hardware Interfaces**

256MB RAM, 1200 MHz Intel, INTEL I3 5TH GEN or AMD processor, keyboard, mouse.

## 4.3. Software Interfaces

System will be provided through a web browser, and it will be compatible with most of the widely used ones, if not all. Thus it will be independent from the operating system of the computer on which it runs.

## 4.4. Communications Interfaces

It is a web application, so constant network communication will be needed to be able to use the system.

**CHAPTER 5**

**NON FUNCTIONAL REQUIREMENTS AND SOFTWARE SYSTEM ATTRIBUTES**

Non-functional Requirements describe quality measures by which a software product must abide. In this section, several non-functional requirements are presented. These requirements cover the performance, security, reliability, availability and maintainability of The Online Collaboration Management System.

Non-Functional Requirements of CollabCall

Non-functional requirements (NFRs) are critical to the success of the CollabCall platform, as they define the quality attributes, performance standards, and usability aspects that the system must achieve. Below are the key non-functional requirements for CollabCall:

1***. Performance Requirements***

* **Response Time:**
  + The platform should respond to user actions (e.g., page loads, task updates) within 2 seconds under normal operating conditions to ensure a smooth user experience.
* **Scalability:**
  + CollabCall must handle a growing number of users and projects without performance degradation. The system should be able to scale horizontally to accommodate up to 10,000 concurrent users.
* Throughput:
  + The platform should support a minimum of 1,000 transactions per minute (TPM) for operations such as task creation, document uploads, and comments.

2**. Reliability and Availability Requirements**

* System Uptime:
  + CollabCall should maintain an uptime of 99.9%, ensuring that the platform is available to users with minimal interruptions.
* Error Handling:
  + The system must provide clear error messages and recovery options in case of failures, maintaining user confidence and minimizing frustration.
* Data Backup:
  + Regular backups should be conducted (at least daily) to prevent data loss and ensure that users can recover their work in case of system failure.

3. Security Requirements

* Data Protection:
  + All user data must be encrypted both in transit (using HTTPS) and at rest to protect sensitive information from unauthorized access.
* User Authentication:
  + CollabCall must implement secure authentication methods, such as OAuth2, to ensure that only authorized users can access their accounts and workspaces.
* Access Control:
  + Role-based access control (RBAC) should be enforced to restrict user permissions based on their roles within projects. This ensures that users can only access data and features relevant to their responsibilities.

4. **Usability Requirements**

* User Interface Design:
  + The UI must be intuitive and user-friendly, allowing users to navigate the platform with minimal training. Users should be able to perform common tasks (e.g., creating workspaces, assigning tasks) within three clicks.
* Help and Support:
  + CollabCall should provide comprehensive help documentation and user support options (e.g., chat support, FAQs) to assist users in resolving issues and optimizing their use of the platform.
* Accessibility:
  + The platform must comply with accessibility standards (e.g., WCAG 2.1), ensuring that users with disabilities can fully interact with all features.

5. Maintainability Requirements

* Code Quality:
  + The codebase should adhere to industry best practices and coding standards, making it easier to maintain and update over time.
* Modularity:
  + The architecture of CollabCall should be modular, allowing for easy addition or modification of features without disrupting existing functionality.
* Documentation:
  + Comprehensive documentation should be provided for both the codebase and user-facing features, facilitating easier onboarding for new developers and users.

6. **Compatibility Requirements**

* Browser Compatibility:
  + CollabCall should be compatible with major web browsers (e.g., Chrome, Firefox, Safari, Edge) and their latest versions to ensure a broad user base can access the platform.
* Device Compatibility:
  + The platform must function optimally across various devices, including desktops, tablets, and smartphones, ensuring a consistent user experience regardless of the device used.

7. Localization Requirements

* Multilingual Support:
  + CollabCall should support multiple languages to cater to a global audience. Users should be able to select their preferred language from the platform settings.
* **Cultural Adaptation:**
  + The platform should consider cultural differences in terms of design, color schemes, and content, enhancing user acceptance and engagement in diverse regions.

**Summary**

The non-functional requirements of CollabCall are essential for delivering a robust, user-friendly, and secure collaborative platform. By addressing these requirements, we ensure that CollabCall not only meets the functional needs of users but also provides a high-quality experience that fosters effective collaboration and engagement in problem-solving initiatives.

## CHAPTER 6 Software Systems Attributes

Software System Attributes of CollabCall

The success of the CollabCall platform hinges on several key software system attributes that define its overall quality, performance, and user experience. These attributes ensure that the platform effectively meets user needs while maintaining high standards of reliability, security, and usability. Below are the primary software system attributes for CollabCall:

**1. Functionality**

**Description:**

Functionality refers to the set of features and capabilities that CollabCall offers to its users.

**Key Aspects:**

* **Core Features:** CollabCall includes essential features such as thread workspace creation, discussion forums, real-time editing, task assignment, and file sharing.
* **Integration Capabilities:** The platform supports integration with third-party tools, enhancing functionality and allowing users to leverage existing workflows.
* **User Interaction:** Users can engage in collaborative problem-solving, communicate effectively, and manage projects seamlessly.

**2. Usability**

**Description:**

Usability focuses on how easy and intuitive the platform is for users to interact with.

**Key Aspects:**

* **User Interface (UI):** CollabCall features a clean and intuitive UI that simplifies navigation and task execution.
* **Learning Curve:** The platform is designed to minimize the learning curve, enabling users to perform tasks efficiently without extensive training.
* **Help and Support:** Comprehensive help documentation and user support options are available to assist users in maximizing their experience.

**3. Reliability**

**Description:**

Reliability measures the platform's ability to perform consistently and without failure over time.

**Key Aspects:**

* **Uptime:** CollabCall aims for an uptime of 99.9%, ensuring that users can access the platform when needed.
* **Error Handling:** The system provides clear error messages and recovery options, maintaining user confidence during failures.
* **Data Integrity:** Robust data management practices ensure the accuracy and consistency of user data.

**4. Performance**

**Description:**

Performance refers to how well the platform responds to user actions and manages resources.

**Key Aspects:**

* **Response Time:** The platform is designed to respond to user actions within 2 seconds under normal operating conditions.
* **Scalability:** CollabCall can accommodate a growing number of users and projects without performance degradation.
* **Throughput:** The system supports a high volume of transactions, ensuring that multiple users can collaborate simultaneously without noticeable lag.

**5. Security**

**Description:**

Security encompasses the measures taken to protect user data and maintain user privacy.

**Key Aspects:**

* **Data Encryption:** All user data is encrypted in transit and at rest to safeguard against unauthorized access.
* **User Authentication:** Secure authentication methods (e.g., OAuth) are employed to verify user identities.
* **Access Control:** Role-based access control (RBAC) ensures that users have appropriate permissions based on their roles within projects.

**6. Maintainability**

**Description:**

Maintainability measures how easily the platform can be updated, modified, and improved over time.

**Key Aspects:**

* **Code Quality:** The codebase adheres to industry best practices, making it easier for developers to understand and modify.
* **Modularity:** The platform's architecture is modular, allowing for the addition or modification of features without disrupting existing functionality.
* **Documentation:** Comprehensive documentation for both developers and users facilitates easier onboarding and ongoing maintenance.

**7. Portability**

**Description:**

Portability refers to the platform's ability to operate across different environments and devices.

**Key Aspects:**

* **Browser Compatibility:** CollabCall is compatible with major web browsers (e.g., Chrome, Firefox, Safari, Edge) to ensure accessibility for a broad user base.
* **Device Compatibility:** The platform functions optimally on various devices, including desktops, tablets, and smartphones, providing a consistent user experience.

**8. Interoperability**

**Description:**

Interoperability measures how well the platform can integrate and communicate with other systems and applications.

**Key Aspects:**

* **Third-Party Integrations:** CollabCall supports integration with popular tools (e.g., Google Drive, Slack, Trello), enhancing its functionality and user experience.
* **API Access:** Developers can create custom integrations through API access, allowing for extended capabilities and workflows.

**Conclusion**

The software system attributes of CollabCall are critical to delivering a robust, user-friendly, and secure collaborative platform. By focusing on functionality, usability, reliability, performance, security, maintainability, portability, and interoperability, we ensure that CollabCall meets the diverse needs of its users while fostering effective collaboration and engagement in problem-solving initiatives. These attributes contribute to the overall quality and success of the platform, making it a valuable tool for individuals and teams alike.

**CHAPTER 7**

***Chapter: Design Constraints of CollabCall***

In the development of CollabCall, a collaborative platform aimed at facilitating community-driven problem-solving, several design constraints emerged. These constraints played a pivotal role in shaping the platform's architecture, user experience, and overall functionality. Understanding these constraints is essential for recognizing the decisions made throughout the design process and the implications they have on the final product.

**1. User Interface (UI) Design Constraints**

**Simplicity and Usability**

One of the primary design constraints for CollabCall was the need for a simple and intuitive user interface. The platform must cater to users with varying levels of technical expertise, from students to professionals. Therefore, we prioritized a clean layout that minimizes cognitive load, allowing users to navigate effortlessly and find the features they need without confusion.

Responsive Design

The increasing use of mobile devices necessitated a responsive design that adapts seamlessly to different screen sizes. CollabCall's layout must ensure that users can access all functionalities, whether they are using a desktop, tablet, or smartphone. This constraint required careful planning in the design of UI components to ensure they scale appropriately across devices.

**2. Accessibility Constraints**

Compliance with Accessibility Standards

Adhering to accessibility standards such as the Web Content Accessibility Guidelines (WCAG) 2.1 was a key design constraint. The platform had to be usable by individuals with disabilities, including those who rely on screen readers or alternative input devices. This requirement influenced design decisions related to color contrast, keyboard navigation, and alternative text for images, ensuring an inclusive experience for all users.

**3. Performance Constraints**

Load Times

Users expect fast load times, which served as a critical design constraint for CollabCall. To meet performance expectations, we needed to optimize our assets (e.g., images, CSS, JavaScript) and implement efficient coding practices. This constraint guided our decisions around how to structure the application and manage resources effectively to minimize delays.

Scalability

As CollabCall is designed to grow with its user base, scalability became a crucial design constraint. The architecture needed to support a large number of concurrent users and projects without performance degradation. This requirement influenced our choice of technologies and the way we structured our database and server interactions.

**4. Security Constraints**

Data Protection Requirements

Given the sensitive nature of user data, security was a paramount design constraint. We established that all user data must be encrypted during transmission and storage to protect against unauthorized access. This requirement shaped our design choices around data flow, storage mechanisms, and user authentication processes.

Role-Based Access Control

The need for role-based access control (RBAC) dictated how we designed user permissions and access levels within the platform. This constraint ensured that users could only access features and information relevant to their roles, minimizing the risk of data breaches and maintaining confidentiality.

**5. Maintainability Constraints**

Code Quality and Documentation

To facilitate ongoing maintenance and future enhancements, the codebase had to adhere to industry best practices. This design constraint emphasized the importance of clean, well-documented code, making it easier for developers to understand, modify, and extend the platform.

Modularity

Designing CollabCall with a modular architecture was essential for maintainability. This constraint allowed us to build features as independent components, enabling easier updates and the addition of new functionalities without disrupting existing ones.

**6. Compatibility Constraints**

Browser and Device Compatibility

To ensure a wide user base, CollabCall had to be compatible with major web browsers (e.g., Chrome, Firefox, Safari, Edge) and function optimally across various devices. This requirement influenced our design decisions, including the choice of technologies and frameworks that support cross-browser functionality.

**7. User Feedback and Iteration Constraints**

User-Centric Design

The design process for CollabCall was guided by a user-centric approach, which required us to incorporate user feedback at various stages. This design constraint meant that we had to remain flexible and open to changes based on user input while balancing the need for timely development and resource allocation.

Conclusion

The design constraints faced during the development of CollabCall were instrumental in shaping the platform. By prioritizing user experience, accessibility, performance, security, maintainability, compatibility, and user feedback, we laid the groundwork for a robust and effective collaborative tool. Understanding these constraints not only highlights the challenges faced but also underscores the strategic decisions made to deliver a high-quality platform that meets the diverse needs of its users. As we move forward, these design constraints will continue to guide future enhancements and iterations of CollabCall, ensuring that it remains a valuable resource for community-driven problem-solving.

CollabCall Project Overview Based on Performance and Functionality Requirements

CollabCall is designed as a robust collaborative platform that facilitates community-driven problem-solving through organized discussions, project management, and resource sharing.

To ensure that CollabCall meets the expectations of its users, we have established specific performance and functionality requirements that guide the development process. This overview outlines the performance requirements that dictate the responsiveness of the platform and the essential functionalities that enhance user experience.

Chapter 8: Performance Requirements

Performance Definition

Performance in the context of CollabCall refers to the responsiveness of the platform during user interactions. It is crucial that the system efficiently manages user loads and maintains optimal response times, ensuring that users can collaborate effectively without interruptions.

Key Performance Metrics

The following performance metrics have been established to ensure a satisfactory user experience:

* **Static Page Response Time:**
  + **90% of responses for static web pages should be within 2 seconds.** This rapid loading time is essential for maintaining user engagement and minimizing frustration.
* **User Operations on Data:**
  + For operations involving data manipulation (e.g., sorting data in a column), response times should be between **5 to 10 seconds** for pages containing up to 100,000 records. This allows users to interact with large datasets without excessive delays.
* **Transaction Response Time:**
  + When executing database transactions (e.g., queries or updates), users should expect a response time of **10 to 20 seconds**. This range accommodates the complexity of the operations while still providing timely feedback.
* **Unacceptable Response Time:**
  + Any response time exceeding **1 minute** is deemed unacceptable and must be addressed to ensure user satisfaction.

Architectural Considerations

To meet these performance requirements, the architecture of CollabCall must incorporate several strategies:

* **Load Balancing:**
  + Implementing load balancing techniques distributes user requests across multiple servers, enhancing reliability and responsiveness as the user base grows.
* **Caching Techniques:**
  + Caching frequently accessed data can significantly reduce response times for static pages and common queries, ensuring that users experience minimal delays.
* **Support for Low Bandwidth:**
  + The design of the platform will ensure that page sizes are optimized for low bandwidth connections, allowing satisfactory response times even in less favorable network conditions.
* **Access Control Mechanism:**
  + Defining an access control mechanism will help manage the number of users consuming various services, ensuring that performance remains stable and predictable.
* **Regular Performance Tuning:**
  + Initiatives such as purging and archiving old data will be implemented to maintain optimal performance and prevent degradation over time.

**Chapter 9: Portal Functionality Requirements**

In addition to the performance metrics, CollabCall is designed to provide a comprehensive set of functionalities that enhance user interaction and collaboration. The following functionalities are essential for the effective operation of the platform:

1. Metadata Synchronization

CollabCall adheres to defined standards for content taxonomy, metadata, and master data management. This requirement ensures that all collaborative content is properly organized and easily retrievable.

2. Full Text Search

The platform includes a full-text search feature that allows users to search for content across the entire portal. This functionality enables users to quickly locate relevant discussions, documents, and resources.

3. Metadata-Based Search

In addition to full-text search capabilities, CollabCall offers metadata-based search functionality. Users can filter their search results using specific metadata attributes, enhancing the precision of their queries.

4. Information Browser

An explorer-type interface provides users with a way to browse all information published on CollabCall. This functionality allows users to discover new content and navigate through various collaborative projects easily.

5. Personalization

CollabCall includes user-specific customization options, such as display themes and homepage customization. This personalization feature allows users to tailor their experience according to their preferences, leading to increased user satisfaction.

6. Portal Usage Reports

The platform generates various reports related to user activity and content engagement. These usage reports help administrators analyze user behavior, identify popular content, and make informed decisions about future enhancements.

7. Self-Service

CollabCall offers self-service functionality for user registration and profile management. Users can create accounts and manage their profiles independently, streamlining the onboarding process and promoting user autonomy.

8. Notifications

Registered users can subscribe to specific content categories. Whenever new content is published in these categories, users receive notifications via email. This feature keeps users informed about relevant updates and encourages ongoing engagement.

9. Portal Administration

CollabCall provides administrative functionalities for user and application management. Administrators can manage user roles, permissions, and overall platform settings to ensure smooth operation and user governance.

**CHAPTER 10**

CollabCall Project: Technology and Platform Comparison

To demonstrate the potential of ***CollabCall*** and highlight the technologies and platforms we chose for its development, we will compare our choices with existing solutions in the market. This not only helps in showcasing our app's potential but also provides a rationale for why we selected specific technologies and approaches based on performance, functionality, and user experience.

1. Backend Technology: Node.js vs. PHP

Our Choice: **Node.js**

* **Reason for Selection:**
  + Node.js is a powerful JavaScript runtime built on Chrome's V8 engine. It allows for non-blocking, event-driven architecture, making it ideal for real-time applications where multiple users are collaborating simultaneously. This capability enhances the responsiveness of CollabCall.
* **Real-World Example:**
  + **Slack:** A real-time messaging platform that uses Node.js for its backend to handle concurrent connections efficiently. This ensures that users can communicate instantly without lag, similar to the real-time collaboration features we aim to implement in CollabCall.

Comparison with: **PHP**

* **Limitations of PHP:**
  + While PHP (e.g., used in platforms like WordPress) is widely utilized for web applications, it traditionally follows a synchronous request-response model which may not handle real-time interactions as effectively as Node.js.

2. Database Management: MongoDB vs. MySQL

Our Choice: **MongoDB**

* **Reason for Selection:**
  + MongoDB is a NoSQL database that provides flexibility in data modeling, accommodating the varied and dynamic nature of user-generated content within CollabCall. Its document-oriented structure allows for quick data retrieval and scalability.
* **Real-World Example:**
  + **Spotify:** Utilizes MongoDB to manage its extensive catalog of music and user data, allowing for flexible querying and quick updates as user preferences change.

Comparison with: **MySQL**

* **Limitations of MySQL:**
  + MySQL is a relational database system that can require predefined schemas, which might not be as adaptable for the evolving requirements of collaborative platforms like CollabCall.

3. Frontend Framework: React vs. Traditional HTML/CSS

Our Choice: **React**

* **Reason for Selection:**
  + React is a JavaScript library for building user interfaces, particularly suited for developing interactive and dynamic single-page applications. It allows for efficient updates and rendering of components, providing a smooth user experience.
* **Real-World Example:**
  + **Facebook:** Uses React extensively to create a dynamic and responsive user interface, ensuring that updates are rendered seamlessly, which is crucial for user engagement.

Comparison with: **Traditional HTML/CSS**

* **Limitations of Traditional Approaches:**
  + While static HTML/CSS is simple and effective for basic pages, it lacks the interactivity and responsiveness that modern web applications require, especially in collaborative environments.

4. Real-Time Communication: WebSocket vs. Polling

Our Choice: **WebSocket**

* **Reason for Selection:**
  + WebSocket provides a full-duplex communication channel over a single, long-lived connection, allowing for real-time data transfer without the overhead of traditional HTTP requests. This is essential for features like live updates in discussion threads and notifications.
* **Real-World Example:**
  + **Trello:** Uses WebSocket technology to enable real-time collaboration, allowing users to see updates from teammates instantly.

Comparison with: **Polling**

* **Limitations of Polling:**
  + Polling involves repeatedly checking the server for updates, which can lead to increased latency and server load, reducing the efficiency of real-time interactions.

5. User Experience and Design: Responsive Design Frameworks

Our Choice: **Bootstrap**

* **Reason for Selection:**
  + Bootstrap is a popular CSS framework that facilitates the development of responsive and mobile-first web applications. By using Bootstrap, CollabCall ensures that users have a consistent experience across devices, from desktops to smartphones.
* **Real-World Example:**
  + **Twitter:** Uses Bootstrap to enhance its platform’s responsiveness, ensuring that users can access their feeds and notifications seamlessly on any device.

Comparison with: **Custom CSS**

* **Limitations of Custom CSS:**
  + While custom CSS allows for unique designs, it can be time-consuming to develop responsive layouts from scratch, which may lead to inconsistencies across different devices if not managed properly.

Conclusion

By selecting technologies like Node.js, MongoDB, React, WebSocket, and Bootstrap, CollabCall is positioned to deliver a highly responsive, scalable, and user-friendly collaborative platform. The comparisons with established platforms such as Slack, Spotify, and Facebook illustrate the potential of our application to meet the demands of modern users. These choices not only enhance performance and functionality but also ensure that CollabCall remains competitive in the collaborative software landscape. This strategic alignment with proven technologies and frameworks underscores our commitment to providing an exceptional user experience and effective collaboration capabilities.

**CHAPTER 11**

# COLLABORATION MANAGEMENT DIAGRAMS

**Flowchart**:  
A diagram of a type comment

Description automatically generated

**11.1 :USE CASE DIAGRAM:**

A use case diagram is a graphic depiction of the interactions among the elements of a system.

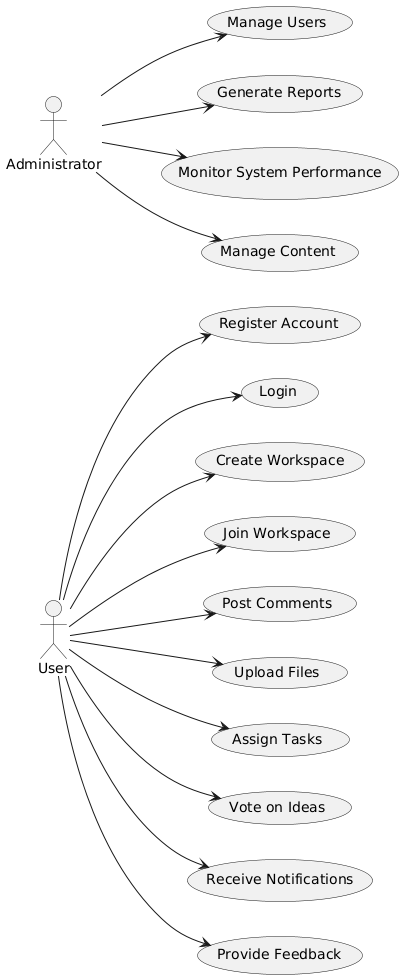
* Description of use case diagram:  
  **Actors:**
  + **User:** Represents any individual interacting with the CollabCall platform.
  + **Administrator:** Represents the individual responsible for managing the platform.
* **Use Cases:**
  + **User Use Cases:**
    - **Register Account:** Allows new users to create an account.
    - **Login:** Enables users to log into their account.
    - **Create Workspace:** Lets users start a new collaborative workspace.
    - **Join Workspace:** Allows users to join existing workspaces.
    - **Post Comments:** Enables users to engage in discussions within workspaces.
    - **Upload Files:** Allows users to share documents and resources.
    - **Assign Tasks:** Enables users to create and assign tasks to team members.
    - **Vote on Ideas:** Allows users to express preferences on proposed ideas.
    - **Receive Notifications:** Keeps users informed of updates and changes.
    - **Provide Feedback:** Enables users to submit their feedback on the platform.
  + **Administrator Use Cases:**
    - **Manage Users:** Allows administrators to oversee user accounts.
    - **Generate Reports:** Lets administrators create reports on platform usage and activity.
    - **Monitor System Performance:** Enables administrators to ensure optimal performance of the platform.
    - **Manage Content:** Allows administrators to oversee and manage published content.

A diagram of a computer

Description automatically generated

[DETAILED VERSION OF USE CASE IS HERE](https://excalidraw.com/DETAILED%20VERSION%20OF%20USE%20CASE)

USER PERSPECTIVE : actor admin relationship



Enquiry

)

(

from use case

user

)

from actor

(

login

(

)

from use case

add details

admin

(

from actor

)

view details

(

from use case

)

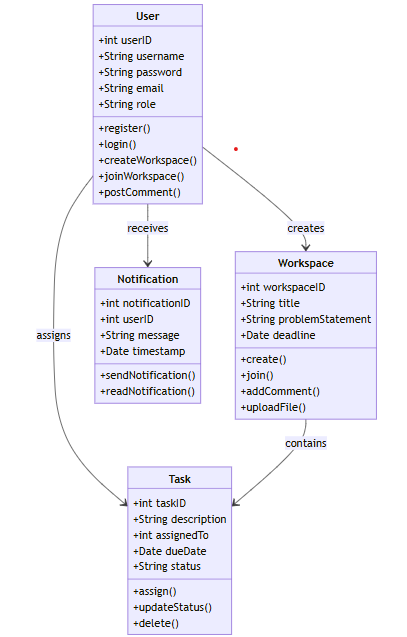
(from use case)

**11.2 :CLASS DIAGRAM:**

A class diagram is an illustration of the relationships and [source code](http://searchsoa.techtarget.com/definition/source-code) dependencies among classes in the Unified Modeling Language (UML). In this context, a [class](http://whatis.techtarget.com/definition/class) defines the [methods](http://searchcio-midmarket.techtarget.com/definition/method) and [variables](http://whatis.techtarget.com/definition/variable) in an [object,](http://searchsoa.techtarget.com/definition/object) which is a specific entity in a program or the unit of code representing that entity. Class diagrams are useful in all forms of object-oriented programming (OOP). The concept is several years old but has been refined as OOP modeling paradigms have evolved.

In a class diagram, the classes are arranged in groups that share common characteristics. A class diagram resembles a [flowchart](http://whatis.techtarget.com/definition/flowchart) in which classes are portrayed as boxes, each box having three rectangles inside. The top rectangle contains the name of the class; the middle rectangle contains the [attributes](http://searchsoa.techtarget.com/definition/attribute) of the class; the lower rectangle contains the methods, also called operations, of the class. Lines, which may have arrows at one or both ends, connect the boxes. These lines define the relationships, also called associations, between the classes.





**11.3 :ACTIVITY DIAGRAM:**

**Activity diagrams** are graphical representations of [workflows](http://en.wikipedia.org/wiki/Workflow) of stepwise activities and actions with support for choice, iteration and concurrency. In the [Unified Modeling Language,](http://en.wikipedia.org/wiki/Unified_Modeling_Language) activity diagrams are intended to model both computational and organisational processes (i.e. workflows). Activity diagrams show the overall flow of control.

A diagram of a workflow

Description automatically generated

**11.4 :SEQUENCE DIAGRAM:**

A **Sequence diagram** is an [interaction diagram](http://en.wikipedia.org/wiki/Interaction_diagram) that shows how processes operate with one another and what is their order. It is a construct of a [Message Sequence Chart.](http://en.wikipedia.org/wiki/Message_Sequence_Chart) A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called **event diagrams** or **event scenarios**.

A sequence diagram shows, as parallel vertical lines (*lifelines*), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner

Object interactions usually begin at the top of a diagram and end at the bottom. In a sequence diagram, object interaction occurs through messages on the vertical and horizontal dimensions and are designated by horizontal arrows and message names. The initial sequence diagram message begins at the top and is located on the diagram's left side. Subsequent messages are added just below previous messages.

Sequence diagram messages may be subdivided by type, based on functionality.

A diagram of a work space

Description automatically generated

**11.5 :COLLABRATION DIAGRAM:**

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among [software](http://searchsoa.techtarget.com/definition/software) [objects](http://searchsoa.techtarget.com/definition/object) in the Unified Modeling Language (UML). The concept is more than a decade old although it has been refined as modeling paradigms have evolved.

A collaboration diagram resembles a [flowchart](http://whatis.techtarget.com/definition/flowchart) that portrays the roles, functionality and behavior of individual objects as well as the overall operation of the system in [real time.](http://searchcio-midmarket.techtarget.com/definition/real-time) Objects are shown as rectangles with naming labels inside. These labels are preceded by colons and may be underlined. The relationships between the objects are shown as lines connecting the rectangles. The [messages](http://whatis.techtarget.com/definition/message) between objects are shown as arrows connecting the relevant rectangles along with labels that define the message sequencing.

A diagram of a flowchart

Description automatically generated



## SNAPSHOTS

Figure 1:Registration Page

Figure 2:Login Page

Figure 3:User Side

Figure 4:Collaboration Time Table

Figure 5 :User Can Register for a meeting

# CONCLUSION

Seeing large number of Collaborations being organized by organizations nowadays, the system can be very helpful in various institutions.The system provides rich support to the Program Committee chairs for managing the Collaboration workflow including customization of Collaboration properties e.g., multiple tracks, deadlines, author submission and reviewer forms, double-blind reviewing, allowing authors to mark conflicts of interest with reviewers, use of external reviewers and meta reviewers. The system also provides filtering, sorting and aggregation functionality as well as emailing capability to authors and reviewers that makes it easy to handle Collaborations with a large number of reviewers and submissions.In the last few years, CMT has been used for more than 2000 Collaborations, workshops, and various other programs.

# REFERENCES

**Bibliography :Collab-Call Project**

1. **React Documentation.** (2023). *React – A JavaScript library for building user interfaces*. Retrieved from [React Official Site](https://reactjs.org/docs/getting-started.html)
2. **Vite Documentation.** (2023). *Vite: Next Generation Frontend Tooling*. Retrieved from [Vite Official Site](https://vitejs.dev/)
3. **Tailwind CSS Documentation.** (2023). *Tailwind CSS: A utility-first CSS framework for creating custom designs*. Retrieved from [Tailwind CSS Official Site](https://tailwindcss.com/docs)
4. **Node.js Documentation.** (2023). *Node.js: JavaScript runtime built on Chrome's V8 JavaScript engine*. Retrieved from [Node.js Official Site](https://nodejs.org/en/docs/)
5. **Express Documentation.** (2023). *Express: Fast, unopinionated, minimalist web framework for Node.js*. Retrieved from [Express Official Site](https://expressjs.com/)
6. **Mozilla Developer Network (MDN).** (2023). *JavaScript Guide*. Retrieved from [MDN JavaScript](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide)
7. **GitHub.** (2023). *Git: A distributed version-control system for tracking changes in source code*. Retrieved from [GitHub](https://github.com/)
8. **W3C.** (2023). *Web Accessibility Guidelines (WCAG)*. Retrieved from [W3C WCAG](https://www.w3.org/WAI/WCAG21/quickref/)
9. **W3C.** (2023). *HTML: HyperText Markup Language*. Retrieved from [W3C HTML](https://www.w3.org/html/)
10. **CSS Tricks.** (2023). *A Complete Guide to Flexbox*. Retrieved from [CSS Tricks](https://css-tricks.com/snippets/css/a-guide-to-flexbox/)
11. **MDN Web Docs.** (2023). *CSS: Cascading Style Sheets*. Retrieved from [MDN CSS](https://developer.mozilla.org/en-US/docs/Web/CSS)
12. **O'Reilly Media.** (2022). *Learning React: Functional Web Development with React and Redux*. Retrieved from [O'Reilly](https://www.oreilly.com/library/view/learning-react/9781492071710/)
13. **Snyk.** (2023). *Open Source Security*. Retrieved from [Snyk](https://snyk.io/" \t "_blank)
14. **Stack Overflow.** (2023). *Community-driven Q&A Platform for Developers*. Retrieved from [Stack Overflow](https://stackoverflow.com/)
15. **Vercel.** (2023). *Vercel: The platform for frontend frameworks and static sites*. Retrieved from [Vercel](https://vercel.com/" \t "_blank)