

# **CONFERENCE MANAGEMENT SYSTEM**

## **23CS5101 – OBJECT ORIENTED PROGRAMMING PROJECT REPORT Batch 2024 (Semester I)**

SUBMITTED BY

TEAM NUMBER - 08

**GARRE DEVIPRATYUSHA**

**2401050069**

**PILLUTLA BEENA JAHANAVI**

**2401050053**

**VELLANKI LAKSHMIPRIYA**

**2401050076**

**PENUMOLU HAMPI**

**2401050117**

**DORADLA NIKHITHA**

**2401050132**

GUIDED BY

**Mr. V. CHANDRA PRAKASH**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**KONERU LAKSHMAIAH DEEMED TO BE UNIVERSITY**

2024-2025

### **DECLARATION**

It is certified that we originally implemented the Project Work titled CONFERENCE MANAGEMENT SYSTEM. No ideas, processes, results, or words of others have been presented as our work. Acknowledgement is given wherever others' work or ideas are utilized.

- a. There is no fabrication of data or results compiled /analyzed.
- b. There is no falsification by manipulating data or processes or changing or omitting data or results.

We understand that the project is liable to be rejected at any stage (even later) if it is discovered that the project has been plagiarized, or significant code has been copied. We understood that if such malpractices are found, the project will be disqualified, and the Degree awarded itself will become invalid.

G. Devi Pratyusha - 2401050069

P. Beena Jahanavi- 2401050053

V. Lakshmipriya– 2401050076

P. Hampi – 2401050117

D. Nikhitha- 2401050132

**KONERU LAKSHMAIAH DEEMED TO BE UNIVERSITY**

VADDESWARAM, GUNTUR – 520 002



**BONAFIDE CERTIFICATE**

“This is to certify that the Project Work report entitled **CONFERENCE MANAGEMENT SYSTEM** carried out by Ms. GARRE DEVIPRATYUSHA (2401050069), Ms. PILLUTLA BEENA JAHANAVI (2401050053), Ms. VELLANKI LAKSHMIPRIYA (2401050076), Ms. PENUMOLU HAMPI (2401050117) Ms. DORADLA NIKHITHA(2401050132) during the academic year 2024 – 2025.

**MR. V. CHANDHRA PRAKASH**

Department of CSE,

KL UNIVERSITY

## **ABSTRACT**

The Conference Management System project is designed to streamline and automate the essential processes involved in organizing and managing professional conferences. It focuses on facilitating efficient interaction between key stakeholders: organizers, participants, and judges. The system allows organizers to send forms, collect submissions, and communicate event details such as the venue, date, and time to participants. Participants can submit their research papers or project details, which are then evaluated by judges through a structured review process. Judges play a critical role in selecting papers and making final award decisions, ensuring a fair and transparent evaluation system. By adopting object-oriented programming principles, the project provides a modular and scalable solution that encapsulates the responsibilities of each role while promoting seamless collaboration. This system addresses common challenges in conference management, such as coordinating submissions, streamlining reviews, and disseminating information, making it a valuable tool for enhancing efficiency and reducing manual effort. The project lays the groundwork for integrating technology into event management and demonstrates how automation can improve logistical workflows, ensuring a professional and productive conference experience.

## **KEYWORDS:**

Conference Management System, GUI (Graphical User Interface), Java, POSTGRESQL, Data Management, Conference

## TABLE OF CONTENTS

CHAPTER NUMBER	TITLE	PAGE NUMBER
	Abstract	IV
	List of Figures	VI
	List of Abbreviations	VII
1	Introduction 1.1 Background 1.2 Problem Statement 1.3 Objectives 1.4 Scope of the project 1.5 Methodology	1 - 4
2	Literature Survey	5 - 7
3	System Analysis 3.1 Proposed System 3.2 Software Requirements Specifications 3.3 Class Diagram 3.4 Use Case Diagrams 3.5 Sequence Diagrams	8 - 18
4	System Design 4.1 System Architecture 4.2 Flow Diagrams & ER Diagrams 4.3 Module Description	19 - 24
5	Implementation Work 5.1 Tools Technologies 5.2 Implementation Details 5.3 Database Design	25 - 28
6	Testing 6.1 Testing Methods 6.2 Test Cases	30 - 31
7	Results & Discussion	32 - 36
8	Conclusion & Future Enhancement	37
9	References	38 - 39
10	Appendices	40 - 56

## LIST OF FIGURES

Figure No	Figure Name	Page Number
1	Waterfall Model	4
2	Class Diagram	13
3 - 9	Use Case Diagrams	13 - 14
10 - 16	Sequence Diagrams	15 - 18
17	System Architecture	19
18 - 24	Data Flow Diagrams	19 - 21
25	Entity Relationship Diagram	22

## LIST OF ABBREVIATIONS

Sl.NO	ABBREVIATIONS
1	<b>CMS</b> – Conference Management System
2	<b>CRUD</b> - Create, Read, Update and Delete operations
3	<b>DBMS</b> - Database Management System
4	<b>SQL</b> - Structured Query Language

## **CHAPTER – 1**

### **INTRODUCTION**

#### **1.1 BACKGROUND**

A Conference Management System plays a vital role in organizing and evaluating participants in academic or professional conferences. Such a system typically involves three primary roles: an organizer, participants, and judges. These roles ensure that the conference process is streamlined, starting from paper submissions to the final awarding of participants. Each role has specific responsibilities that contribute to the success of the event, from logistical planning to evaluating the quality of submitted work and recognizing outstanding contributions.

The organizer manages the logistics of the conference by handling tasks such as sending forms to potential participants and collecting their submissions. Additionally, the organizer communicates essential details like the venue, date, and time of the event to participants whose submissions are accepted. This role ensures that the event runs smoothly by facilitating effective communication between all parties involved and providing a framework for the conference's operation.

Participants are the individuals who submit their research papers or projects for consideration in the conference. Their submissions include key details such as their name, contact information, and the title of their work. Participants represent the core contributors to the conference, as their work forms the basis for review and recognition. Their submissions are typically reviewed by judges to determine their quality and relevance to the conference's theme or objectives.

Judges are responsible for evaluating the submitted work. The evaluation process is often divided into stages. In the initial stage, judges review the submissions and decide which ones qualify for presentation at the conference. Once a submission is selected, the judges may proceed to a second stage to determine if the work merits an award or special recognition. This process ensures that only the most deserving contributions are highlighted, promoting excellence and innovation within the conference's domain.



## **1.2 PROBLEM STATEMENT**

The Conference Management System streamlines the process of organizing and evaluating research submissions for a conference. It involves three main roles: organizers, participants, and judges. The organizer's responsibility is to distribute submission forms to participants and share event details, such as the venue, date, and time, with those whose papers are selected. Participants submit their research papers for consideration. Judges review these papers to decide which ones qualify for presentation and determine if participants should receive awards based on their contributions. The system follows a sequential process, starting with the organizer sending forms to participants, followed by participants submitting their papers. Judges then evaluate the submissions, and the organizer communicates event details to selected participants. A final judge makes the award decisions. This system ensures a structured and efficient workflow for managing research paper submissions, evaluations, and awards in a conference setting.

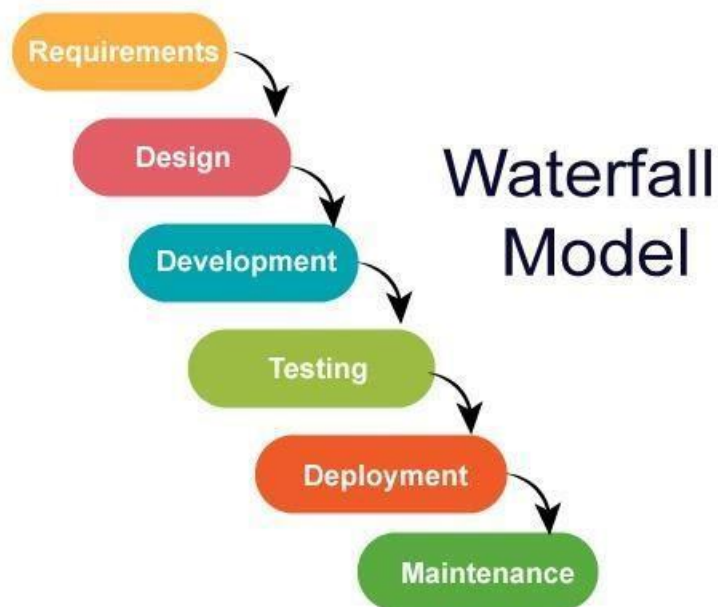
## **1.3 OBJECTIVES**

The objective of the Conference Management System is to create an efficient and structured platform for managing research submissions, evaluations, and communications in a conference setting. The system aims to streamline the submission process by enabling organizers to distribute forms and collect papers seamlessly. It provides a structured mechanism for judges to review and evaluate submissions, ensuring that only high-quality research is selected. Clear communication is prioritized, allowing organizers to share essential event details with participants whose papers are chosen. Additionally, the system supports informed decision-making by final judges when awarding participants for their contributions. By ensuring traceability, simplifying participant interaction, and promoting a modular design, the system offers transparency, scalability, and ease of use. Ultimately, it replicates real-world conference workflows to enhance efficiency and maintain professionalism.

## 1.4 SCOPE OF THE PROJECT

The scope of the Conference Management System encompasses the automation and streamlining of tasks involved in managing research submissions, evaluations, and communications for academic or professional conferences. The system is designed to meet the needs of organizers, participants, and judges, ensuring a structured workflow from paper submission to award finalization. Organizers can distribute submission forms, track submissions, and communicate event details, such as venue, date, and time, to selected participants. Participants can submit their research papers and receive updates on their status while judges can review submissions to determine their suitability for presentation and finalize award decisions. The system is tailored for small to medium-sized conferences, with potential for future expansion to support multiple users, advanced scoring mechanisms, and multi-round evaluations. This project ensures efficiency, transparency, and professionalism in the conference management process.

## 1.5 METHODOLOGY



- **REQUIREMENT GATHERING AND ANALYSIS** – all possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **SYSTEM DESIGN** – the requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- **IMPLEMENTATION** – with inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- **INTEGRATION AND TESTING**– All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **DEPLOYMENT OF SYSTEM** – Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- **MAINTANANCE** – There are some issues which come up in the client environment. To fix those issues, patches are released. Also, to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

## **CHAPTER - 2**

### **LITERATURE REVIEW**

#### **[1] Evolution of Conference Management Systems**

The development of conference management systems (CMS) has not only streamlined event logistics but also redefined participant experiences. Early CMS were primarily used for simple administrative tasks such as attendee registration and schedule planning. However, modern CMS now include features such as integrated payment gateways, automated email reminders, and real-time updates, improving both operational efficiency and attendee satisfaction. Recent studies also highlight the growing trend of integrating machine learning algorithms into CMS to predict attendance trends, optimize session allocations, and personalize the event experience for participants. Furthermore, the adoption of mobile applications linked to CMS has enhanced accessibility, enabling users to manage schedules, view session details, and network on the go. Despite these advancements, challenges such as data security and system scalability remain critical, especially for large-scale international events. Future research calls for more robust CMS architectures to address these challenges while providing a seamless experience for organizers and participants alike.

#### **[2] Artificial Intelligence in Event Management**

AI-driven solutions in event management are paving the way for more intelligent, automated, and personalized experiences. One area of significant advancement is in the automation of paper review processes. AI algorithms, particularly those utilizing natural language processing, can analyze submissions for relevance, originality, and quality, significantly reducing the workload on human reviewers. Additionally, AI enhances participant experiences through chatbots that provide real-time event assistance and recommendation systems that suggest sessions based on individual interests. Studies also explore how AI can predict and analyze attendee engagement using metrics such as session attendance and interaction levels. However, while AI offers numerous advantages, challenges such as algorithm transparency, ethical concerns, and the potential for biased decision-making remain. Researchers suggest that incorporating human oversight in AI systems can mitigate these challenges while maximizing efficiency and fairness in event management..

### **[3] Hybrid Conferences: Benefits and Challenges**

The emergence of hybrid conferences has fundamentally transformed how events are conducted, offering participants the flexibility to choose between physical and virtual attendance. Research indicates that hybrid formats foster inclusivity by accommodating diverse participants, particularly those who face financial, geographical, or health-related barriers. Technological tools such as virtual networking platforms, live polls, and interactive Q&A sessions enhance engagement for remote attendees, ensuring they feel equally involved. However, hybrid conferences also introduce complexities for organizers, including the need for high-quality streaming services, effective time zone management, and synchronized interactions between on-site and online participants. Studies further highlight the psychological challenges for remote attendees, such as feelings of isolation or disengagement compared to their in-person counterparts. To address these issues, researchers recommend innovative solutions such as virtual reality (VR) environments to simulate in-person experiences and AI-driven analytics to measure and improve engagement.

### **[4] Peer-Review Models in Academic Conferences**

Peer-review models are critical to maintaining the academic rigor and credibility of conference proceedings. Traditional blind review processes have long been the standard, but recent studies reveal their vulnerability to implicit biases based on factors such as institutional affiliation or geographic location. Double-blind reviews, where both author and reviewer identities are concealed, are often considered more equitable but can be logistically challenging to implement. Open peer reviews, on the other hand, promote transparency and accountability by revealing reviewer identities and feedback, though they risk deterring honest critique due to potential conflicts of interest. Recent advancements in technology, such as AI-assisted review tools, are transforming the peer-review process by offering automated suggestions and flagging inconsistencies in submissions. However, the integration of AI into peer review requires careful oversight to avoid over-reliance on algorithms and ensure that human expertise remains central. Future research suggests a hybrid model that combines traditional review processes with AI support for optimal outcomes.

## **[5] Sustainability Practices in Conferences**

Sustainability in conference management is becoming a top priority as the global focus on environmental conservation intensifies. Studies emphasize that virtual and hybrid conferences offer significant environmental benefits by reducing travel-related carbon emissions. Moreover, the adoption of digital solutions, such as online registration systems and e-programs, reduces paper waste, making events more eco-friendly. Organizers are also exploring innovative practices like carbon offset programs, reusable materials for in-person events, and sustainable catering options. However, researchers caution against underestimating the environmental impact of virtual events, particularly in terms of energy consumption for data streaming and server usage. Future studies recommend adopting renewable energy sources for digital infrastructure and incorporating green practices into all stages of event planning. These findings highlight the need for a comprehensive approach to sustainability that balances technological and environmental considerations.

## **[6] Digital Transformation in Academic Events**

The digital transformation of academic events has created new opportunities for knowledge dissemination, networking, and collaboration. Virtual and hybrid conferences allow researchers from around the world to share insights without the logistical constraints of physical travel. Advanced features such as virtual poster sessions, breakout rooms, and AI-driven matchmaking systems enhance interaction and foster meaningful connections among participants. Mobile applications linked to conference platforms further improve engagement by providing real-time updates, session reminders, and opportunities for instant networking. However, the rapid shift to digital formats has not been without challenges. Studies highlight issues such as "Zoom fatigue," reduced opportunities for informal networking, and the risk of technical disruptions during key sessions. To address these concerns, researchers suggest combining synchronous and asynchronous elements to provide flexibility, incorporating gamification to boost engagement, and ensuring robust technical support. As digital transformation continues, its success will depend on the ability to balance technological innovation with the core objectives of academic events.

## **CHAPTER – 3**

### **SYSTEM ANALYSIS**

#### **3.1 PROPOSED SYSTEM**

The proposed Conference Management System is designed to streamline and automate the key processes involved in organizing academic or professional conferences. It aims to facilitate smooth interaction between organizers, participants, and judges while ensuring transparency and efficiency throughout the event. The system enables organizers to send submission forms and event details to participants, while participants can easily submit their research papers and receive status updates. Judges can review submissions, select papers, and finalize award decisions. With clear workflows and a modular design, the system ensures that tasks are performed systematically, with logs tracking all actions for transparency. Additionally, the system is scalable, allowing for future enhancements such as multi-round evaluations and real-time notifications. Ultimately, the proposed system will improve the overall efficiency and organization of conferences, making the process more user-friendly for all stakeholders involved.

#### **ADVANTAGES:**

- Increased efficiency.
- Transparency.
- Reduced errors.
- User friendly.
- Scalability.
- Automated Award Decisions.
- Cost-Effective.
- Streamlined communications.

## **3.2 SOFTWARE REQUIREMENTS SPECIFICATIONS(SRS)**

### **3.2.1 FUNCTIONAL REQUIREMENTS**

#### **➤ ORGANISER FEATURES**

**Description:** The system must enable the organizer to send paper submission forms to participants. These forms will include all necessary details, such as the paper submission deadline, submission guidelines, and the format required for submission.

**Functional Requirements:**

- The organizer can enter participant details (name, email) to send forms.
- Forms must include clear instructions on how to submit papers, required fields, and submission deadlines.
- The system must send confirmation emails to participants once the form is dispatched.

#### **➤ PARTICIPENT FEATURES**

**Description:** Participants must be able to submit their research papers or abstracts for review by the judges. This feature allows participants to upload their papers and provide necessary details like the paper title, author(s) name, email, and any other required information.

**Functional Requirements:**

- Participants can upload their paper in a predefined format (e.g., PDF, Word document).
- The submission form will require the participant to fill in basic information such as the title of the paper, abstract, and keywords.
- The system will validate the submission to ensure all required fields are filled out before allowing the participant to submit the paper.
- Once submitted, the system will confirm the successful submission and send a confirmation email to the participant.

#### **➤ JUDGE FEATURES**

**Description:** Judges are responsible for reviewing the papers submitted by participants, providing their evaluation and feedback based on



predefined criteria (e.g., paper quality, originality, relevance).

Functional Requirements:

- Judges can access all the papers submitted for review in the system.
- The system must allow judges to view detailed information about each paper, including the paper title, author details, abstract, and the full paper.
- Judges can rate each paper on various criteria (e.g., content quality, writing style, relevance to conference theme) and provide detailed comments and feedback.
- Judges should be able to submit their reviews for each paper, indicating whether the paper should be accepted, rejected, or revised.

➤ **AWARD DECISION :**

Description: Before the award decision is made, the system must have predefined award categories and criteria to ensure fair evaluation. Categories could include:

- **Best Paper:** Awarded to the paper that demonstrates exceptional quality, originality, and contribution to the field.
- **Best Presentation:** Awarded to the participant who presents their paper most effectively at the conference.
- **Most Innovative Paper:** Recognizing the paper with the most novel or groundbreaking ideas.
- **Best Student Paper:** Specifically for student participants, recognizing excellence in academic research.

Functional Requirements:

- The system must allow judges to choose from predefined award categories.
- The system will display the criteria for each category, so judges can evaluate the papers based on clear guidelines.
- The system may allow multiple awards to be given in different categories, depending on the number of submissions and the judging process.

➤ **Final Award Decision by Senior Judges or Organizer**

Description: Once the aggregated ratings and reviews are submitted, the final decision for the award(s) is made. This can either be done by a senior judge or a panel of senior judges based on the aggregated scores, or by the conference organizer if applicable.

Functional Requirements:

- The system will allow the final decision-maker to review the aggregated scores, comments, and rankings for each paper.
- If necessary, the final decision-maker can override or adjust the aggregated scores based on additional factors (e.g., alignment with conference goals, importance of paper to the field, or presentation quality).
- The final decision-maker may also have the option to create a shortlist of papers that have been ranked the highest, from which the final awardees will be chosen.

## **COMMUNICATION AND NOTIFICATION**

Description: The Communication and Notification system is a vital feature that ensures seamless interaction between participants, organizers, and judges. It helps in maintaining transparency, keeping all stakeholders informed, and ensuring that key activities, such as form submissions, paper reviews, and award decisions, are communicated in a timely and effective manner.

Functional Requirements:

- **Participant:** The system should notify participants when their paper has been reviewed, including the feedback, score (if applicable), and any revision requests. If their paper has been selected for the next stage (e.g., for presentation), this

should be indicated.

- **Organizer:** The system should notify the organizer about the review status, including whether the paper has been approved or rejected by the judges.

### 3. Event Details Notification

- **Description:** Once the paper is accepted, the organizer must send event details, including the venue, date, and time, to the participant.
- **Functional Requirements:**
- **Participant:** The system will send an email with the conference details, such as venue, date, time, and any instructions for attending (e.g., virtual conference links, deadlines for paper submission, etc.).
- The system should also remind participants about the event as the date approaches (e.g., a week or a day before the event).

## LOGGING AND TRACABILITY

Description: Logging and Traceability are essential components for ensuring the integrity, transparency, and accountability of the conference management system. These features allow administrators, organizers, and even participants to track the history of actions and decisions throughout the entire conference process. From paper submission to award decision, the system must provide detailed logs and the ability to trace every action taken.

Functional Requirements:

- Log events such as paper submissions, form submissions, and updates to personal information.
- Log events when the organizer sends event details, tracks paper submissions, or finalizes award decisions.
- Log when judges review papers, submit ratings, or finalize decisions regarding a paper's status (e.g., approved or rejected).

➤ **USER MANAGEMENT**

Description: The system should support multiple roles, each with specific permissions.

Roles determine what a user can see and do in the system. By assigning different roles, the system can enforce role-based access control, ensuring that users only have access to the features they need.

Functional Requirements:

- Can submit papers, track the status of their submissions, and receive event details. They should not have access to judging decisions or event management functions .
- Can manage event details, send forms, track paper submissions, and view reviews. Organizers can also send communications to participants and finalize decisions.

### **3.2.2 NON – FUNCTIONAL REQUIREMENTS**

➤ **PERFORMANCE REQUIREMENTS**

- The system should process requests (such as paper submission or reviewing papers) within 2-3 seconds.
- The system should be able to handle high traffic, especially during peak submission or review times. It must support simultaneous users (participants, organizers, judges) without degrading performance.

➤ **SECURITY REQUIREMENTS**

- Secure login mechanism, including encrypted passwords and the option for two-factor authentication (2FA) for administrators and organizers.
- Role-based access control (RBAC) should be implemented, ensuring that users only have access to the features they are authorized to use .

➤ **USABILITY REQUIREMENTS**

- The system should provide a user-friendly interface for non-technical users.
- The system must comply with web accessibility standards to ensure it is usable for users with disabilities.

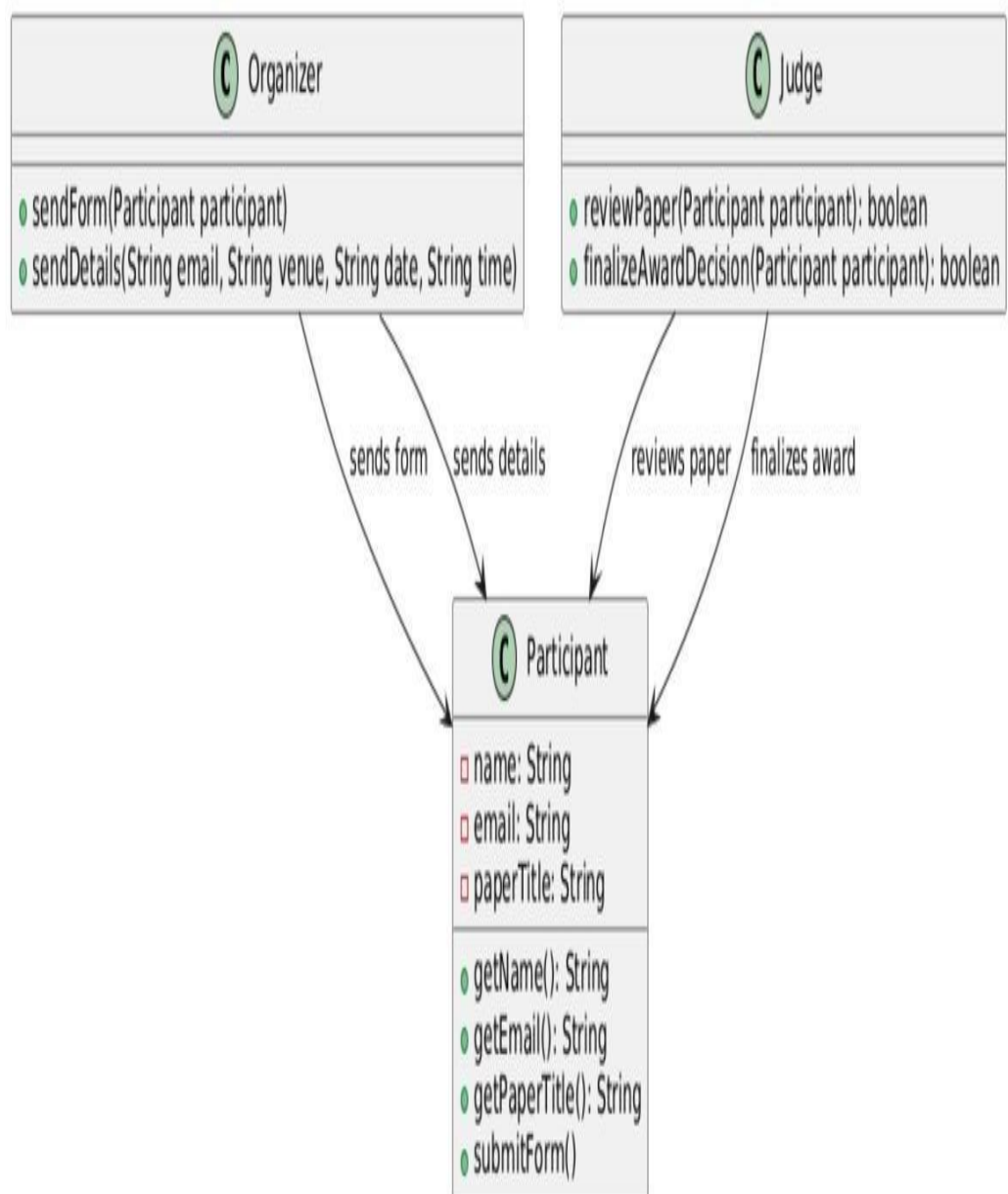
➤ **RELIABILITY REQUIREMENTS**

- The system should have 99.9% uptime during operational hours.
- Data integrity
- must be maintained, and no data should be lost during system failures.

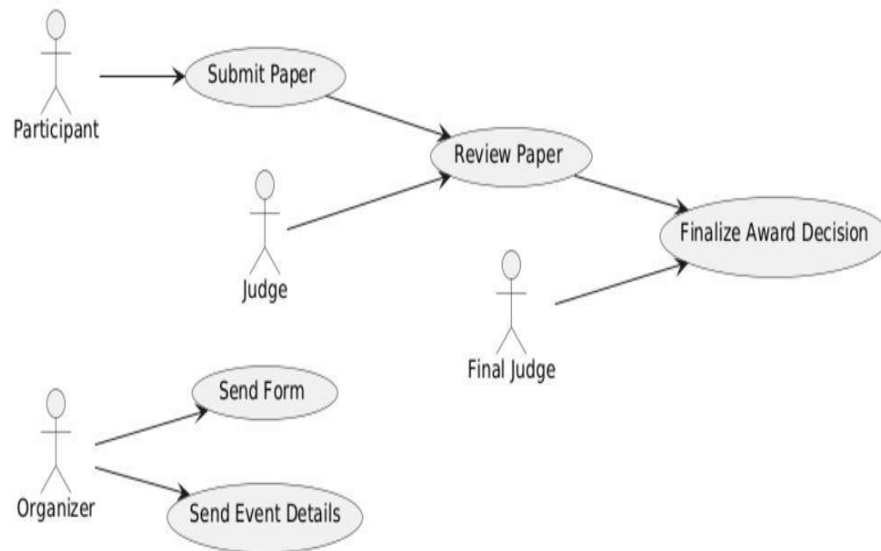
➤ **COMPATABILITY REQUIREMENTS**

- The system should work on all modern browsers (e.g., Chrome, Firefox, Safari, Edge).
- The system should be responsive, adapting to various screen sizes, and work seamlessly on both desktop and mobile devices.

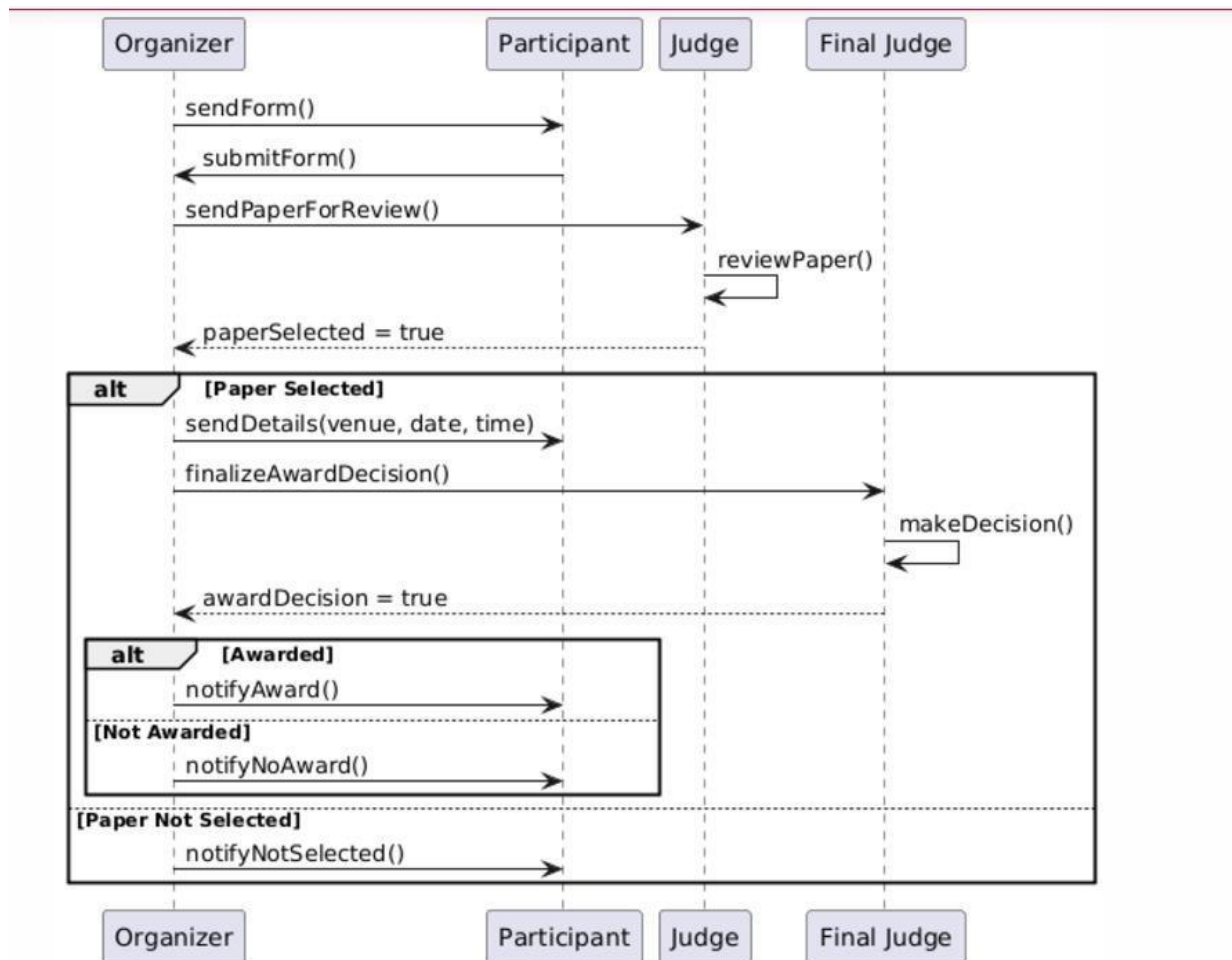
### 3.3 CLASS DIAGRAM



### 3.4 USE CASE DIAGRAMS



### 3.4 SEQUENCE DIAGRAMS

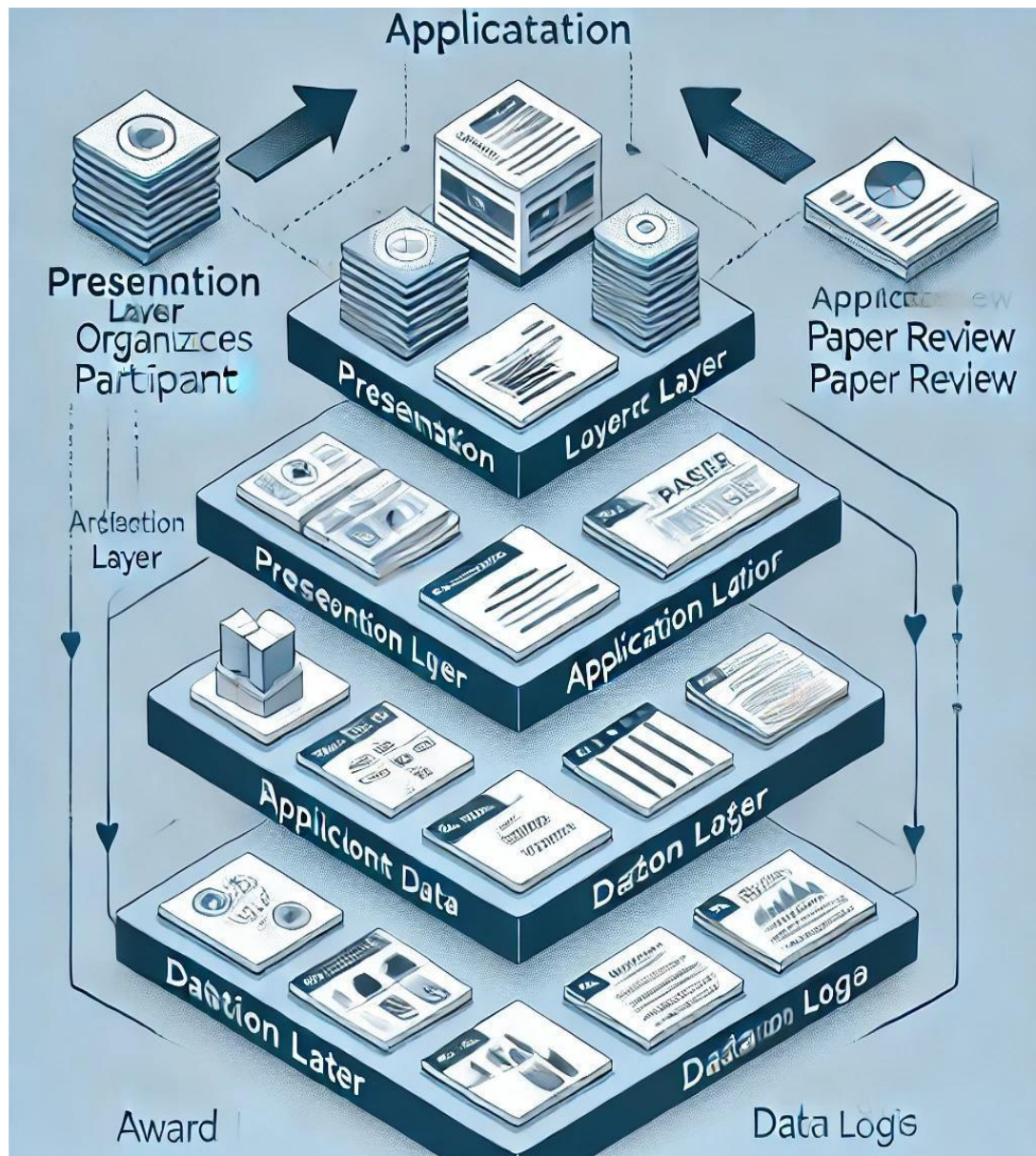




## CHAPTER – 4

### SYSTEM DESIGN

#### 4.1 SYSTEM ARCHITECTURE



## **4.1 FLOW DIAGRAMS**

### **4.1.1 DATA FLOW DIAGRAM**

The Data Flow Diagram (DFD) illustrates how data moves between various components, processes, and data stores within the system. Below, we describe Level 1 and Level 2 DFDs for the Conference Management System.

Processes Explained:

Process 1: Send Form

External Entity: Organizer

Description: The organizer sends the form to the Participant to fill in paper details. This is the first step where the organizer initiates the process.

Process 2: Submit Form

External Entity: Participant

Description: The Participant submits their form, which contains the paper title and possibly other required details (e.g., abstract, authors). The system receives this data and stores it in the Data Store (Paper Details).

Process 3: Review Paper

External Entity: Judge

Description: The Judge reviews the paper submitted by the participant. This process involves reviewing the content of the paper, evaluating its relevance, and deciding if the paper will be accepted for the conference.

Process 4: Send Event Details

External Entity: Organizer

Description: If the paper is accepted by the judge, the organizer sends the event details (venue, time, date) to the Participant. These details are stored in the Data Store (Participant Info).

Process 5: Finalize Award

External Entity: Final Judge

Description: The Final Judge makes the decision on whether the participant will be awarded for

the paper or not. This process concludes with a final decision, which is communicated to the participant.

#### Data Flows:

**Submit Form:** Data flows from the Participant to the Conference System (contains paper details).

**Review Paper:** The Judge reviews the paper details stored in the Data Store (Paper Details) and decides whether the paper is selected.

**Send Event Details:** If the paper is selected, the Organizer sends the event details (venue, date, time) to the Participant.

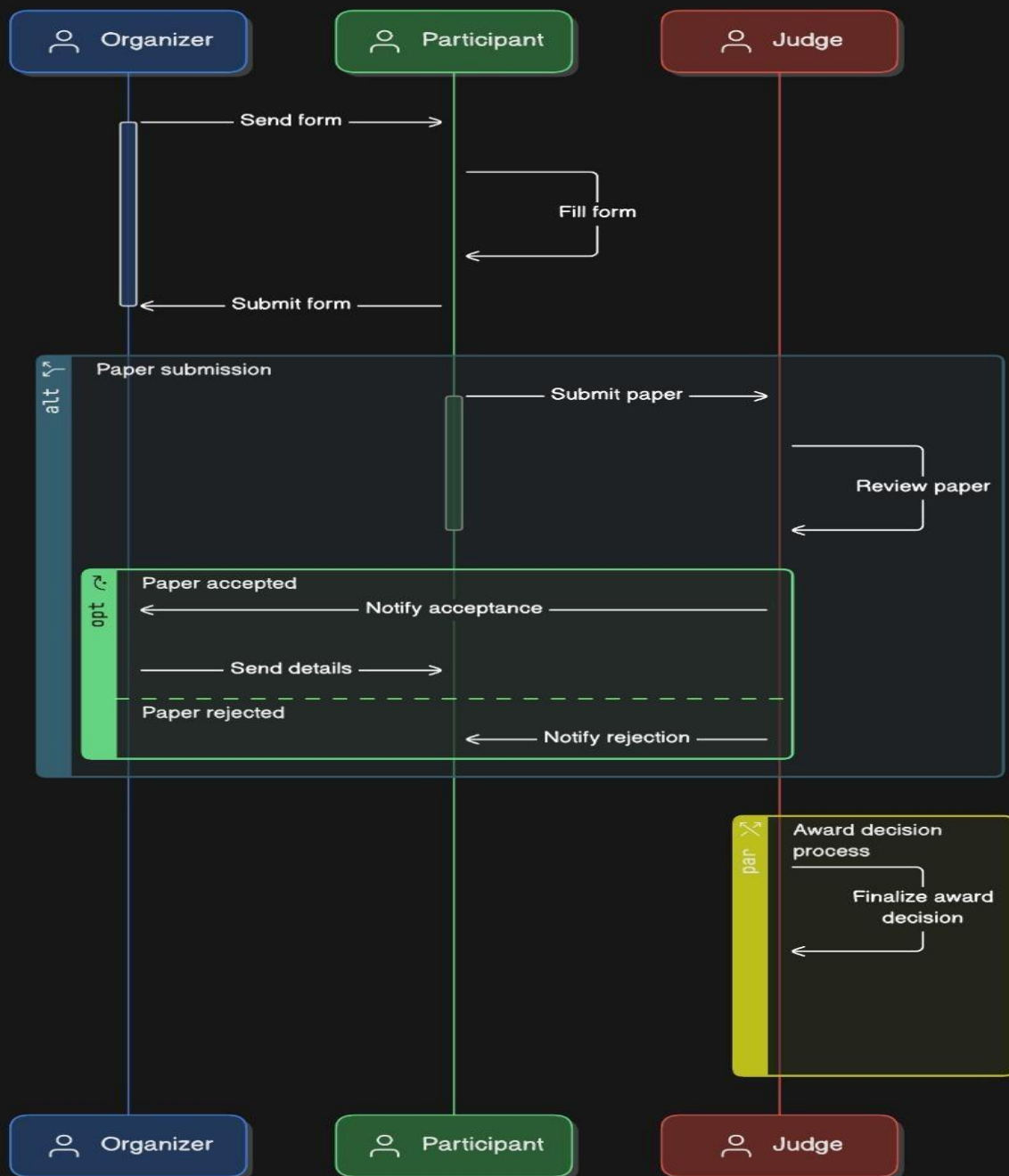
**Finalize Award:** The Final Judge accesses the Data Store (Participant Info) to make the final decision on whether the Participant gets an award.

#### Data Stores:

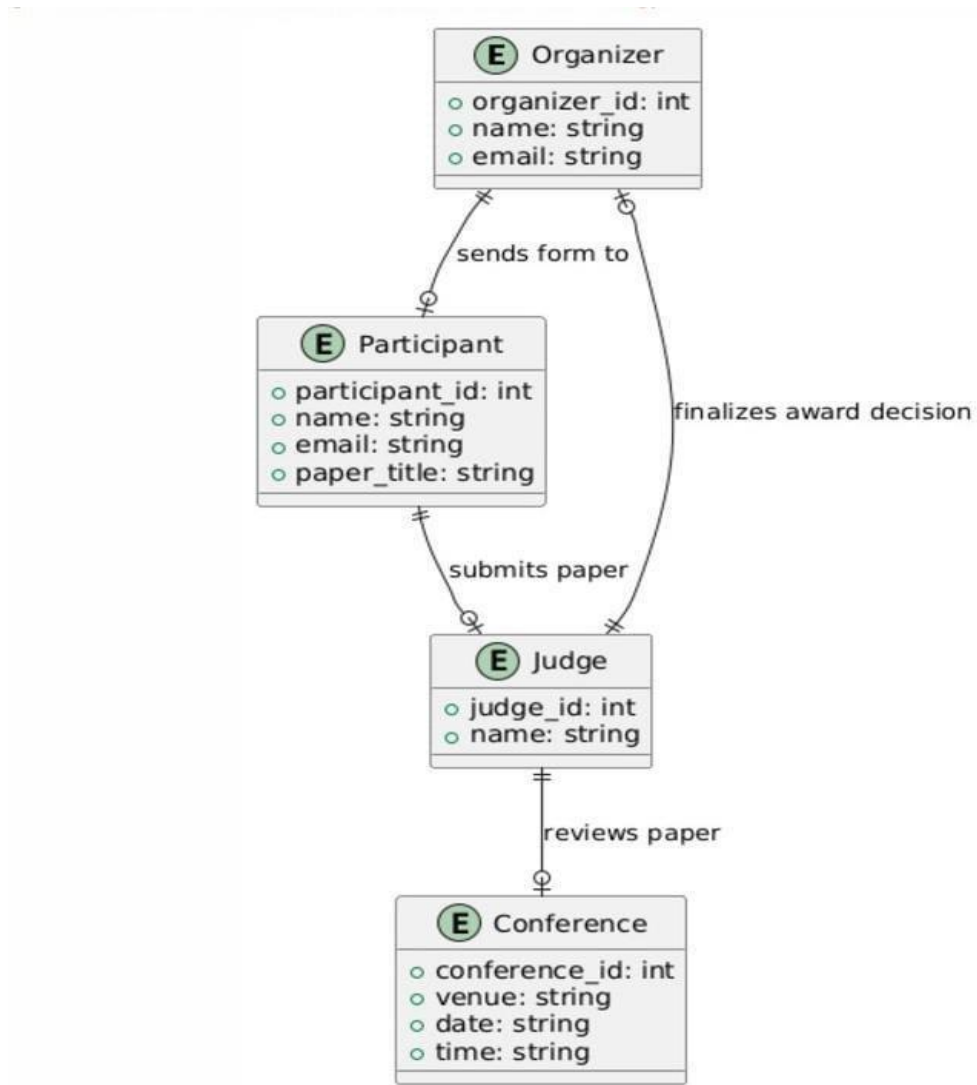
**Paper Details:** Stores all the details related to the paper submitted by the participant (e.g., paper title, abstract).

**Participant Info:** Stores all the details about the participant, such as name, email, and event details once selected.

## Conference Paper Submission Process



### 4.1.2 ER DIAGRAM



### 4.2 MODULE DESCRIPTION

#### ORGANISER MODULE

The Organizer Module is responsible for managing the overall conference process, including coordinating with participants and judges. The organizer's primary tasks involve sending paper submission forms to participants, ensuring they have all the necessary details to submit their papers. Once submissions are received, the organizer tracks the status of each paper, including whether it has been reviewed and selected.

## **PARTICIPANT MODULE**

The Participant Module allows participants to submit their papers and interact with the conference system. Participants receive the paper submission form from the organizer, which contains all the necessary information for submitting their paper. Once the form is completed, the participant submits their paper, including the title and other relevant details.

## **JUDGE MODULE**

The Judge Module is designed to manage the paper review and award decision processes. Judges are responsible for reviewing the papers submitted by participants, evaluating them based on criteria such as originality, content quality, and relevance to the conference theme. After reviewing the papers, judges decide whether the paper should be selected for the event. If a paper is selected, the judge provides feedback, and the paper moves forward in the evaluation process. The final judge is tasked with making the award decision, determining whether the participant should receive an award based on the overall quality of the paper. This module enables judges to assess and provide crucial input that helps in selecting the best papers for the conference.

## **CONFERENCE MODULE**

The Conference Module serves as the central hub for managing and coordinating the entire conference process. It orchestrates the interactions between the organizer, participants, and judges, ensuring that all steps—from paper submission to award decisions—are carried out efficiently. The module initiates the workflow by enabling the organizer to send paper submission forms to participants, tracks the progress of submissions, and facilitates communication between the judges and participants. It also ensures that event details, such as the venue, date, and time, are sent to selected participants. Additionally, the Conference Module handles the review process, allowing judges to assess the papers and make final award decisions. By managing all essential processes in a unified platform, the Conference Module ensures a seamless and organized conference experience.

## **USER INTERACTION**

The interaction begins with the organizer sending paper submission forms to participants, who can easily complete and submit their papers through the system. Participants are notified about the status of their submissions and, if selected, receive event details such as venue, date, and time. Judges interact with the system by reviewing submitted papers, providing feedback, and making final decisions regarding awards.

## **PAPER REVIEW PROOCCESS**

Once a participant submits their paper, it enters the review phase, where judges assess its quality based on criteria such as originality, relevance, and academic rigor. The judges provide feedback on the paper and determine whether it should be selected for the conference. This process guarantees that only the best papers are showcased at the conference, and it ensures transparency and fairness in the selection and evaluation procedures.

## **AWARD DECISION PROCESS**

The Award Decision Process is the final step in the Conference Management System, where judges determine whether a participant should receive an award based on the quality of their paper. After the paper has undergone thorough review by the judges, the final decision is made by the lead or final judge, who evaluates the feedback and assessments from earlier rounds. The decision considers factors such as the paper's originality, contribution to the field, and alignment with the conference's objectives.

## **EVENT DETAILS NOTIFICATION**

The Event Details Notification is an essential feature of the Conference Management System, ensuring that selected participants receive timely and accurate information about the conference. Once a participant's paper is selected for presentation, the organizer sends a notification containing crucial event details, including the conference venue, date, and time. This notification helps participants prepare for the event by providing them with all the logistical information they need.

## **CHAPTER – 5**

### **IMPLEMENTATION**

#### **5.1 TOOLS AND TECHNOLOGIES**

##### **Hardware Specifications:**

Processor : Intel Core i5 (or) AMD Ryzen 5 (or better)  
RAM : 8 GB (minimum), 16 GB (recommended)  
Hard Disk : 250 GB and Above

##### **Software Specifications:**

Operating System : Windows, MacOS, Linux  
Server-side Script : Java, SQL  
IDE : Visual studio or Eclipse  
GUI : Java script, HTML, CSS  
Libraries Used : Spring boot, JDBC, Junit.

#### **5.2 IMPLEMENTATION DETAILS**

Steps for Developing the Conference Management System

##### **1. Requirements Gathering**

- Identify and gather requirements from the key stakeholders like conference organizers, participants, judges, and sponsors.

##### **2. System Design**

- Design the database schema to store data such as user information, paper submissions, reviews, and event details.

##### **3. UI Development**

- Create wireframes and prototypes of the system interface for each user role (organizer, participant, judge). Consider usability, accessibility, and responsiveness for the UI.



#### 4. **Development Planning**

- Divide the development into smaller tasks and sprints (if following Agile). Define the scope of each sprint and prioritize tasks.

#### 5. **Integration**

- Ensure smooth integration between the back-end logic (e.g., paper submission, review management) and the front-end interface.

#### 6. **Testing**

- Test the user interface for functionality, usability, and responsiveness across different browsers and devices.

#### 7. **Deployment**

- Set up the production database and migrate data from development to production.

#### 8. **Ongoing Maintenance**

- After deployment, monitor the system for any issues or bugs and fix them promptly.

This structured approach ensures clarity and organization throughout the development process of the Conference Management System.

### **5.3 DATABASE DESIGN**

The database design for the conference management system aims to effectively store and manage all key data related to the conference process, including participants, judges, paper submissions, event details, reviews, and award decisions. The design includes multiple interconnected tables, each representing a different entity such as Organizer, Participant, Paper, Judge, Review, Event Details, and Award. By establishing clear relationships such as one-to-many and many-to-one between these tables, the design ensures data integrity and simplifies querying for conference-related information. The use of normalization principles up to 3NF ensures efficient storage and retrieval of data while eliminating redundancy. Additionally, foreign key constraints and indexes are implemented to maintain referential integrity and optimize query performance. Overall, this structured database design provides a solid foundation for managing the various components of a conference management system and can be easily extended to accommodate future needs.

```

Port [5432]:
Username [postgres]:
Password for user postgres:

psql (16.4)
WARNING: Console code page (437) differs from Windows code page (1252)
        8-bit characters might not work correctly. See psql reference
        page "Notes for Windows users" for details.
Type "help" for help.

postgres=# \c conference;
You are now connected to database "conference" as user "postgres".
conference=# \dt;
           List of relations
 Schema |      Name      | Type  | Owner
-----+-----+-----+-----
 public | participants   | table | postgres
(1 row)

conference=# SELECT * FROM participants;
 id |      name      |      email      |      paper_title
-----+-----+-----+-----
  1 | Alice Johnson | alice@example.com | The Future of AI
  4 | Ramya         | ramya@example.com | The Future of Cybersecurity
  5 | geetha        | geetha@example.com | data science
(3 rows)

conference=# |

```

## CHAPTER – 6

### TESTING

#### 6.1 TESTING METHODS

##### ➤ UNIT TESTING

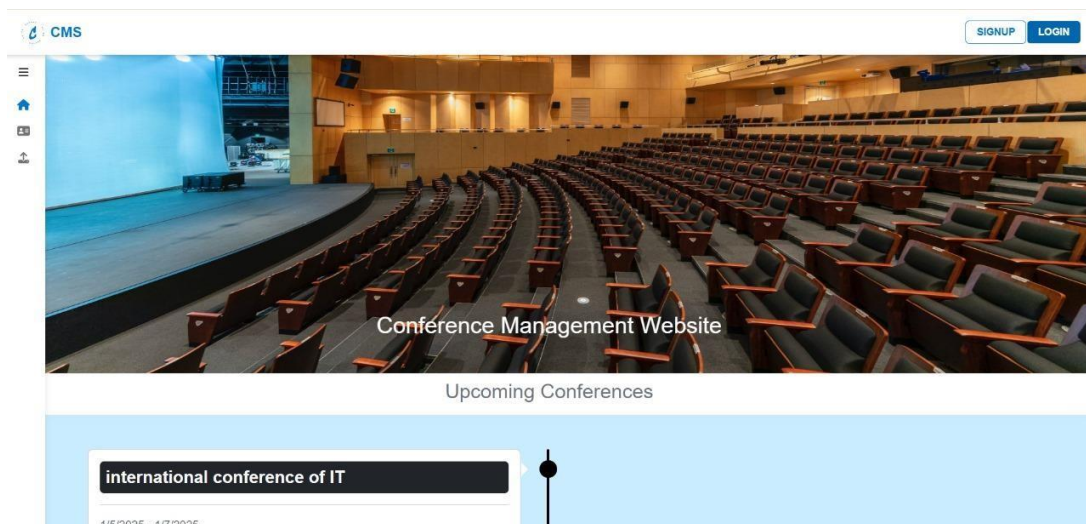
- Unit tests for the Conference System class can simulate different workflows, ensuring that the entire process (from form submission to award decision) functions as intended. Using a unit testing framework like JUnit in Java, developers can automate these tests to quickly identify and fix bugs, improve code quality, and ensure the stability of the system during future changes or enhancements.

##### ➤ INTEGRATION TESTING

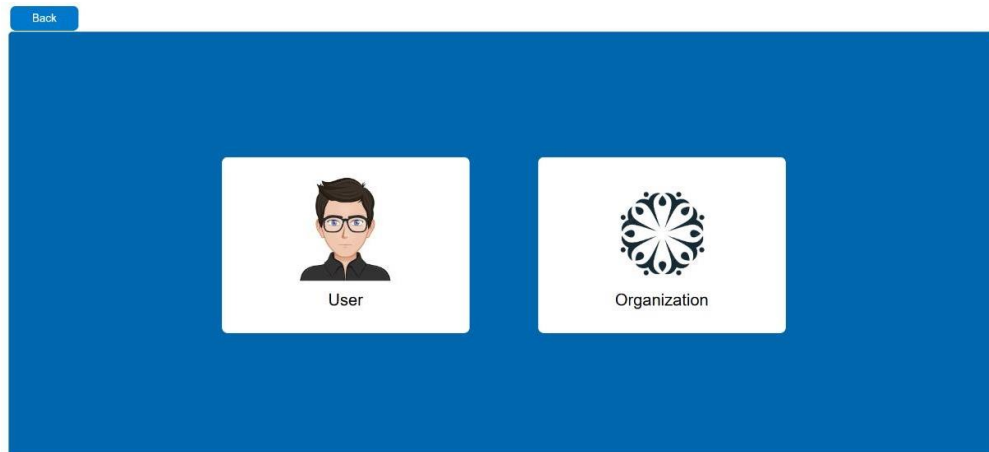
Integration testing also helps detect issues that may not be apparent in unit testing, such as data flow errors, miscommunication between modules, and failures in the system's business logic when components interact. Tools like JUnit with Mockito for mocking dependencies or Selenium for end-to-end testing can be used to automate the integration tests, ensuring the modules are well-integrated, reliable, and perform as expected under real-world conditions.

#### 6.2 TEST CASES

##### Case – 1:



## Case – 2:



## Case – 3:

### The best platform for Conferences

This website is a platform for organizing and managing events. It includes features like online registration, speaker and agenda management, and communication tools.

This platform streamlines the organization of conferences and makes it easier for organizers to deliver a successful event.

Already have an account, then click on the below Login button!

Login

### Signup

Username:

Email address:

Role:  
☐ Publisher ☐ Attendee

Password:

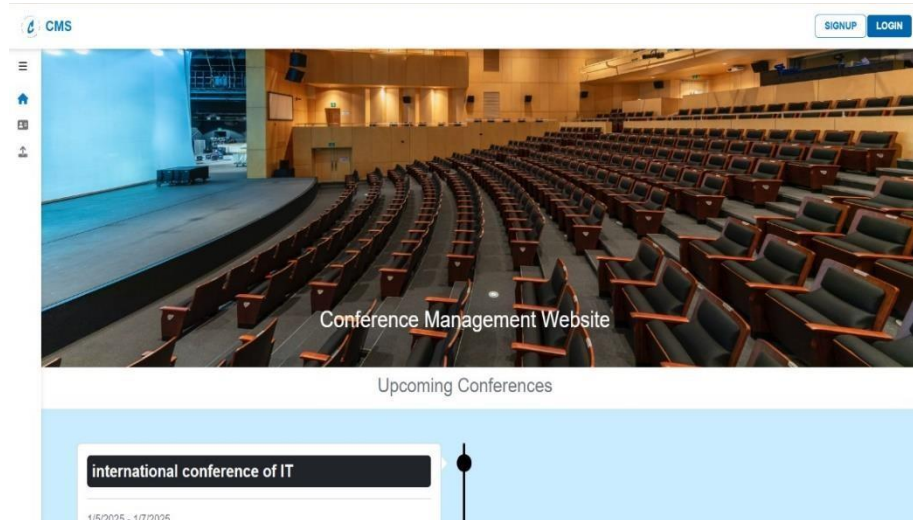
Confirm Password:

Submit

## Case – 4:

The image shows a web form for 'Organization Signup'. It is split into two panels. The left panel has a blue background and contains the text: 'The best platform for Conferences', 'This website is a platform for organizing and managing events. It includes features like online registration, speaker and agenda management, and communication tools.', 'This platform streamlines the organization of conferences and makes it easier for organizers to deliver a successful event.', and 'Already have an account, then click on the below Login button!' with a 'Login' button. The right panel has a white background and contains the title 'Organization Signup' and four input fields: 'Organization name:', 'Email address:', 'Password:', and 'Confirm Password:'. A 'Submit' button is at the bottom of the right panel.

## RESULTS



## The best platform for Conferences

This website is a platform for organizing and managing events. It includes features like online registration, speaker and agenda management, and communication tools.

This platform streamlines the organization of conferences and makes it easier for organizers to deliver a successful event.

Don't have an account, please SignUp first!

Signup

### Organization Login

Email address:

jackson3@gmail.com

Password:

\*\*\*\*\*

Submit

[Forgot Password ?](#)

## The best platform for Conferences

This website is a platform for organizing and managing events. It includes features like online registration, speaker and agenda management, and communication tools.

This platform streamlines the organization of conferences and makes it easier for organizers to deliver a successful event.

Don't have an account, please SignUp first!

Signup

### Login

Email address:

alience@gmail.com

Password:

\*\*\*\*\*

Role:

☒ Publisher

☐ Attendee

Submit

[Forgot Password ?](#)

## **CHAPTER – 8**

### **CONCLUSION & FUTURE SCOPE**

#### **8.1 CONCLUSION**

The Conference Management System (CMS) developed serves as a comprehensive solution to streamline the complex processes involved in organizing academic conferences. Through the integration of key modules—Organizer, Participant, Judge, and the central Conference System—the system ensures smooth coordination between all stakeholders, from paper submission to award decisions. By automating tasks such as form distribution, paper review, and event notifications, the system reduces manual work and enhances efficiency, offering a seamless experience for both organizers and participants. The overall user experience, contributing to more successful and well-organized conferences. The solution is adaptable and can be expanded with additional features, ensuring its relevance for future events.

#### **8.2 FUTURE SCOPE**

The Future Scope of the Conference Management System presents numerous opportunities for expansion and improvement. Enhancements such as a more interactive Graphical User Interface (GUI), AI-driven paper evaluation tools, and real-time communication features could significantly improve user experience and operational efficiency. Integrating advanced reporting and analytics will provide organizers with valuable insights into paper submissions and reviewer feedback. The development of a mobile app and cloud integration would ensure greater accessibility and scalability. Additionally, features like plagiarism detection, integration with academic platforms, and multi-language support would further streamline the process for global participants. With these improvements, the system could serve a wider range of users, making it a more powerful tool for academic events and conferences.

## REFERENCES

- [1] Design and Implementation of a Collaborative Conference Management System" (IEEE Xplore) This paper explores a collaborative conference management system focusing on enhancing interaction among authors, reviewers, and organizers. It highlights system design principles and scalability.  
Link: <https://ieeexplore.ieee.org/document/4536947>
- [2] The Transformative Evolution of Conference Management Systems" (Journal of Digital System Development) Discusses the historical and technological evolution of conference management systems, detailing key features like centralized submissions and review processes while forecasting future trends in CMS development.  
Link: <https://e-journal.uum.edu.my/index.php/jdsd/article/view/22842>
- [3] Microsoft Conference Management Toolkit (CMT) Highlights features like flexible submission and review workflows, support for double-blind reviewing, and automated conflict detection for academic peer reviews.  
Link: <http://research.microsoft.com/docs/help/overview/CMT.html>
- [4] Framework and Development of CMS  
A detailed framework discussing CMS modules like login, paper submission, reviewer management, and event scheduling. The article highlights the use of evolutionary design methodologies for CMS, focusing on usability, reducing paperwork, and improving administrative efficiency.
- [5] Designing Collaborative CMS Systems  
This study uses WebML to design multi-role CMS platforms. It emphasizes efficient collaboration between users like authors and reviewers and offers insights into reducing manual work for editors. This approach highlights the potential of automation and adaptive design.
- [6] Comprehensive Guide to Conference Management" (vFairs) :This guide offers insights into different types of conferences, such as entrepreneurial, sustainability-focused, and cultural events. It explains planning processes like setting objectives, managing budgets,



and coordinating logistics.

[7] Event and Conference Planning Handbook" (Capterra)

Highlights essential software tools and their impact on efficient event management, including automated registration, attendee tracking, and feedback collection.

[8] Conference Management Best Practices" (Eventbrite)

Details methodologies for enhancing attendee engagement, managing digital marketing campaigns, and leveraging event apps to streamline the experience.

[9] IEEE Xplore: Conference Management Studies

Scholarly articles exploring innovations in conference management systems, including AI-driven tools and hybrid event solutions .

[10] Google Scholar - Academic Papers on Event Management

A variety of peer-reviewed articles on conference systems, covering topics such as participant satisfaction, technology integration, and cost optimization.

## APPENDICES

### SOURCE CODE :

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class ConferenceSystem {
    public static void main(String[] args) {
        // Initialize Organizer, Participant, and Judge
        Organizer organizer = new Organizer();
        Participant participant = new Participant("Ramya", "ramya@example.com", "The Future
of Cybersecurity");
        Judge judge1 = new Judge();
        Judge judge2 = new Judge();

        // Step 1: Organizer sends form to participant
        organizer.sendForm(participant);

        // Step 2: Participant submits the filled form with paper details
        participant.submitForm();

        // Step 3: First judge reviews and selects the paper
        boolean isSelected = judge1.reviewPaper(participant);
        if (isSelected) {
            System.out.println("Judge 1: Paper selected!");

            // Step 4: Organizer sends venue, date, and time details
            String venue = "Conference Hall A";
            String date = "2024-11-15";
            String time = "11:00 AM";
            organizer.sendDetails(participant.getEmail(), venue, date, time);

            // Step 5: Final judge decides if the participant receives an award
            boolean isAwarded = judge2.finalizeAwardDecision(participant);
            if (isAwarded) {
                System.out.println("Final Judge: " + participant.getName() + " has been awarded!");
            } else {
                System.out.println("Final Judge: " + participant.getName() + " did not receive the
award.");
            }
        } else {
            System.out.println("Judge 1: Paper not selected.");
        }
    }
}

// Organizer class
static class Organizer {
```

```

public void sendForm(Participant participant) {
    System.out.println("Organizer: Form sent to " + participant.getName());
}

public void sendDetails(String email, String venue, String date, String time) {
    System.out.println("Organizer: Sending details to " + email);
    System.out.println("Venue: " + venue);
    System.out.println("Date: " + date);
    System.out.println("Time: " + time);
}
}

// Participant class
static class Participant {
    private String name;
    private String email;
    private String paperTitle;

    public Participant(String name, String email, String paperTitle) {
        this.name = name;
        this.email = email;
        this.paperTitle = paperTitle;
        saveToDatabase();
    }

    public String getName() {
        return name;
    }

    public String getEmail() {
        return email;
    }

    public String getPaperTitle() {
        return paperTitle;
    }

    public void submitForm() {
        System.out.println("Participant " + name + " submitted the paper titled: " + paperTitle);
    }

    private void saveToDatabase() {
        String url = "jdbc:postgresql://localhost:5432/conference";
        String user = "postgres"; // Replace with your PostgreSQL username
        String password = "nikhitha@123"; // Replace with your PostgreSQL password

        try (Connection connection = DriverManager.getConnection(url, user, password)) {
            String sql = "INSERT INTO participants (name, email, paper_title) VALUES (?, ?, ?)";
            PreparedStatement statement = connection.prepareStatement(sql);

```

```

        statement.setString(1, this.name);
        statement.setString(2, this.email);
        statement.setString(3, this.paperTitle);

        statement.executeUpdate();
        System.out.println("Participant " + this.name + " saved to the database.");
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
}

// Judge class
static class Judge {
    public boolean reviewPaper(Participant participant) {
        System.out.println("Judge: Reviewing paper titled '" + participant.getPaperTitle() + "' by "
+ participant.getName());
        // Simulate a paper selection process
        return true; // assuming the paper is selected for now
    }

    public boolean finalizeAwardDecision(Participant participant) {
        System.out.println("Final Judge: Finalizing decision for " + participant.getName());
        // Simulate the final award decision process
        return true; // assuming the participant is awarded
    }
}

```

## **SUPPORTING DOCUMENTATION**

### **INSTALLATION GUIDE**

This section outlines the steps to install PostgreSQL and set up the database for the Orphanage Management System.

### **STEPS TO INSTALL POSTGRESQL ON WINDOWS**

#### **1) Download PostgreSQL Installer:**

- Visit the official PostgreSQL download page: [PostgreSQL Downloads](#).
- Select "Windows" and choose the Interactive Installer by EnterpriseDB for the latest version.

#### **2) Run the Installer:**

- Double-click the downloaded .exe file to start the installation wizard.
- Click Next on the welcome screen.

#### **3) Choose Installation Directory:**

- Specify the installation directory or leave it as default. Click Next.

#### **4) Select Components:**

- Choose components to install:
  - PostgreSQL Server
  - pgAdmin 4 (GUI management tool)
  - Optional: Command Line Tools
  - Click Next.

#### **5) Data Directory:**

- Set the data directory where PostgreSQL will store its data files or accept the default location. Click Next.

#### **6) Set Password for Superuser:**

- Enter a password for the database superuser (default is postgres). Make sure to remember this password. Click Next.

7) Port Configuration:

- Leave the default port (5432) unless you have another application using it. Click Next.

8) Locale Settings:

- Choose the default locale or leave it as is. Click Next.

9) Review Installation Summary:

- Review the installation settings and click Next to begin installation.

10) Finish Installation:

- Once installation completes, uncheck "Launch Stack Builder at exit" and click Finish.

11) Launch pgAdmin 4:

- Open pgAdmin 4 from your Start Menu.
- In pgAdmin, connect to your server by entering the password set during installation.

12) Verify Installation

- Open a command prompt and type `psql -U postgres`. Enter your password when prompted.
- Run `SELECT version ();` to confirm that