FYP COMPLETION CERTIFICATE

Student Name: Muhammad Bilal Raza Enrolment No: 01-131202-022

Student Name: Shahzaib Ali Enrolment No: 01-131202-029

Programme of Study: Bachelor of Software Engineering

Project Title: BibConfetti

It is to certify that the above students' project has been completed to my satisfaction and to my belief, its standard is appropriate for submission for evaluation. I have also conducted plagiarism test of this thesis using HEC prescribed software and found similarity index at _____ that is within the permissible limit set by the HEC. I have also found the thesis in a format recognized by the department.

Supervisor's Signature: _____

Date: _____ Name: Engr. Muhammad Waleed Khan

CERTIFICATE OF ORIGINALITY

This is certify that the intellectual contents of the project "BibConfetti" are the product of my/our own work except, as cited properly and accurately in the acknowledgements and references, the material taken from such sources as research journals, books, internet, etc. solely to support, elaborate, compare, extend and/or implement the earlier work. Further, this work has not been submitted by me/us previously for any degree, nor it shall be submitted by me/us in the future for obtaining any degree from this University, or any other university or institution. The incorrectness of this information, if proved at any stage, shall authorities the University to cancel my/our degree.

Name of the Student: Muhammad Bilal Raza

Signature: _____ Date: _____

Name of the Student: Shahzaib Ali

Signature: _____ Date: _____

PROJECT TITLE (MENTION PROJECT TITLE HERE)

Sustainable Development Goals

(Please tick the relevant SDG(s) linked with FYDP)

SDG No	Description of SDG	SDG No	Description of SDG
SDG 1	No Poverty	SDG 9✓	Industry, Innovation, and Infrastructure
SDG 2	Zero Hunger	SDG 10	Reduced Inequalities
SDG 3	Good Health and Well Being	SDG 11	Sustainable Cities and Communities
SDG 4✓	Quality Education	SDG 12	Responsible Consumption and Production
SDG 5	Gender Equality	SDG 13	Climate Change
SDG 6	Clean Water and Sanitation	SDG 14	Life Below Water
SDG 7	Affordable and Clean Energy	SDG 15	Life on Land
SDG 8	Decent Work and Economic Growth	SDG 16	Peace, Justice and Strong Institutions
		SDG 17	Partnerships for the Goals

Range of Complex Problem Solving			
	Attribute	**Complex Problem**	
1	Range of conflicting requirements	Involve wide-ranging or conflicting technical, engineering and other issues.	✓
2	Depth of analysis required	Have no obvious solution and require abstract thinking, originality in analysis to formulate suitable models.	
3	Depth of knowledge required	Requires research-based knowledge much of which is at, or informed by, the forefront of the professional discipline and which allows a fundamentals-based, first principles analytical approach.	
4	Familiarity of issues	Involve infrequently encountered issues	✓
5	Extent of applicable codes	Are outside problems encompassed by standards and codes of practice for professional engineering.	
6	Extent of stakeholder involvement and level of conflicting requirements	Involve diverse groups of stakeholders with widely varying needs.	✓
7	Consequences	Have significant consequences in a range of contexts.	
8	Interdependence	Are high level problems including many component parts or sub-problems	
Range of Complex Problem Activities			
	Attribute	**Complex Activities**	
1	Range of resources	Involve the use of diverse resources (and for this purpose, resources include people, money, equipment, materials, information and technologies).	
2	Level of interaction	Require resolution of significant problems arising from interactions between wide ranging and conflicting technical, engineering or other issues.	✓
3	Innovation	Involve creative use of engineering principles and research-based knowledge in novel ways.	
4	Consequences to society and the environment	Have significant consequences in a range of contexts, characterized by difficulty of prediction and mitigation.	
5	Familiarity	Can extend beyond previous experiences by applying principles-based approaches.	

Abstract

The Academic Conference & Research Article Management System named BibConfetti is a web-based software designed to streamline and enhance the management of academic conferences and research article submissions. Its primary purpose is to facilitate the end-to-end process of organizing conferences, managing various types of publications, and ensuring a seamless user experience for researchers, conference organizers, Reviewing Committees, and administrators. The Academic Conference & Article Management System aims to provide a comprehensive solution for the management of academic conferences, research article submissions, and the publication process. It uses AI, NLP Models for automation which help to assign reviewers and for reviewers to give their feedback.

Keywords: Academic conference, research article management, web-based software, conference organization, automation, reviewer assignment, submission management, key insights of publications.

Dedication

First, we dedicate this project to Almighty God, our Creator, our strong pillar, our source of inspiration, wisdom, knowledge and understanding. He has been the source of our strength throughout this FYP project, and it is only on his wings that we have soared. Second, we would like to dedicate our project work to our parents and teachers.

A special feeling of gratitude goes to our loving parents and teachers, whose words of encouragement and striving for perseverance ring in our ears. We also dedicate this great project to our friends who supported us throughout the process. We faced obstacles on the way towards the completion of a project, but our parent's and teachers'

encouragement and prayer never let us lose hope we kept working hard day and night and eventually we achieved the goal.

Acknowledgments

We thank Allah Almighty a lot, being the best arranger, for all the duties of this project. Undoubtedly, I would like to thank my parents and my teachers as well for the moral support that they have always given me plus the great range of learning opportunities that Bahria University (BSEAS) Islamabad has provided. Last but not the least. I am indebted to our supervisor Engr. Muhammad Waleed Khan for the leadership provided, and I will never forget the kindness and grace given to me by my friends and family. Primarily, I would like to thank to all people who were involved in this project BibConfetti, it will never came true without you.

Contents

Table of Figures

List of Tables

Chapter 1
Introduction

This chapter provides the introduction of our application BibConfetti. This chapter will highlight the motivation, problem statement, contribution, and objectives of the project. BibConfetti is a web application through which user a web-based conference and research article software system with a better user interface, integrated plagiarism checker and AI-NLP based large language models for ensuring factual correctness and summarizing the theme of the research article submitted.

1.1 Motivation

The main motivation behind this project is to enhance The Academic Conference & Research Article Management System represents a new, self-contained product designed to revolutionize the management of academic conferences and research article submissions. It is not a replacement for existing systems but rather a standalone solution catering to the specific needs of academic institutions and researchers. The system operates independently but may interface with external services for plagiarism checking and payment processing. While it functions autonomously, its seamless integration with external systems ensures a holistic approach to conference management and publication workflows.

1.2 Objectives

To this end the main objectives of Project are:

1. *Auto Assignment of reviewers.*
2. *Key insights for helping reviewer for feedback.*
3. *Chat with document*
4. *Plagiarism Checker Integration*
5. *Pricing Plans and Packages*
6. *Better UI for understanding of users*

1.3 Main contributions

Our tool is unique from other that firstly we are using AI features for automation and this application also provide one to all solution to its users as it consists of integrated plagiarism checker, payment module which other such tools does not have such integrated modules users have to switch to other applications for payment and plagiarism purposes. The Academic Conference & Research Article Management System is a web-based software designed to streamline and enhance the management of academic conferences and research article submissions. Its primary purpose is to facilitate the end-to-end process of organizing conferences, managing various types of publications, and ensuring a seamless user experience for researchers, conference organizers, reviewers, and administrators. The system's benefits include improved collaboration among researchers, faster and more transparent review processes, and enhanced organization and scheduling of academic conferences.

1.4 Report organisation.

Chapter 1:

In this chapter, we have given a short introduction to our project and have discussed some objectives and uniqueness of our project BibConfetti.

Chapter 2:

In this chapter, we discussed how our project is different from other conference management tools and publications tools and what are advantages and disadvantages of our project.

Chapter 3:

In this chapter, we discussed use case diagram and basic flow of our project and discussed some functional and non-functional requirements. Understanding of different actors involve by using use case descriptions.

Chapter 4:

In this chapter, we discuss UML diagrams like class, activity, sequence diagrams for better understanding of project.

Chapter 5:

In this chapter, we discussed technologies and different types of tools which we used for creation of our application.

Chapter 6:

In this chapter, we test correctness of our project features to verify that if all requirements are fulfilling or not and also perform different types of testing so that to make sure that quality of system is maintained.

Chapter 7:

In this chapter, we conclude whole project and learning from this project and future work relates to this project.

Chapter 2

Background Study/Literature Review

2.1 Rationale of BibConfetti

BibConfetti was created and is being developed based on the main issues of current Academic Conference Management Systems and the related advantages offered by AI and automation technologies. Through the delivery of a useful interface, functioning artificial intelligence and powerful security measures, BibConfetti purpose is to facilitate online conferences, application, feedback, publication, and as a result, the efficiency and effectiveness of academic conferences will improve.

2.1.1 Challenges and Opportunities

This study shows that although computerized systems of academic conference management have shown great progress, a several of the challenges are still being experienced. In this category, there are questions regarding user interaction, growth restriction, security, as well as manual control of the approach to reviewing. Also, as the enormous volume of the submissions and the prompt decision-making capacity required have seen come to fore as major concerns for conference organizers and reviewers. Nevertheless, these difficulties can give rise to the inventive application of artificial intelligence (AI) technology and automation and thus may be regarded as the beginning of a new period of automation that possibly leads to new highs in productivity.

2.1.2 Gap Analysis

While the academic conference management systems which are already in the market deliver substantial functionalities, there are a lot of gaps in this sector that BibConfetti wants to address. For instance, the systems have to feature improved user experience, easily integrate AI-driven functions, and provide support for teamwork/communication among the concerned bodies. Through technology adoption, especially the one that emerging technologies are offering, and by addressing these gaps, BibConfetti endeavors to move beyond the scope of existing solutions and create a comprehensive and functional platform for academic conferences' administration.

2.1.3 Theoretical Framework

The processes of BibConfetti design and development are steered by information systems theory, human-computer interaction (HCI) and scholarly communication science Ideas of system usability, user-centered design, and social aspects of technology adoption educate the team making BibConfetti a user-friendly application that meets the expectations of its users.

2.1.4 Emerging Trends and Innovation

The latest AI, NLP and machine learning advancements bring various benefits for improvement of the academic conference systems in terms of their efficiency and effectiveness. These AI-powered tools can accomplish tasks like reviewer assignment, plagiarism checking, and content analysis with less effort on the users' side. Besides that, data analytics tools may translate into finding out submission trends, reviewer behavior, and publication impact.

2.2 Existing Solution and Technologies

The two famous applications which are currently providing these services are Microsoft Conference Management Toolkit and EasyChair however our application, BibConfetti which is an Academic Conference & Research Article Management System represents a new, self-contained product designed to revolutionize the management of academic conferences and research article submissions. It is not a replacement for existing systems but rather a standalone solution catering to the specific needs of academic institutions and researchers. The system operates independently but may interface with external services for plagiarism checking and payment processing. While it functions autonomously, its seamless integration with external systems ensures a holistic approach to conference management and publication workflows.

https://easychair.org/

https://cmt3.research.microsoft.com/

2.2.1 Drawbacks: These are some points gathered through feedback of these application users which should be improved: MS-CMT has been criticized for having a somewhat outdated and less friendly UI. Bugs in CMT can disrupt the submission

and review process. Users have expressed concerns about data privacy and security when using Easy Chair. Navigating its features and settings can take time.

2.3 Literature Review

In this section, we have briefly explained different literature that we studied related to the technologies and modules related to our system

2.3.1 A Framework for Conference Management System

This study shows that most universities and colleges use one of the several web-based conference management systems that have been built under the open source paradigm. Every conference management system created used web servers. The process of the event planning requires much paperwork and takes a lot of time. This encompasses the announcement of the call for papers, author submissions, review process, author registration, and overall participant registration. Thus, an easy to use framework is required for the conference management system to satisfy the needs of non-technical users. [1]

2.3.2 Survey of Conference Management Systems

This article examines the several conference management systems that are used to organize conferences across the globe. Every system has certain features that are specific to it as well as certain shared features. Approaches for the specification of functions, behavior, and communication are further separated into the external interaction specification approaches. We summarize how the systems' functionalities are used after surveying them. Under the following headings, we have conducted surveys on EDAS, Confious, OpenConf, ConfTool, and PaperDyne: 1) System 2) Conferences 3) TPC 4) Reviewers5) Papers 6) Reports and 7) Notifications. [2]

2.3.3 A Topic-Based Reviewer Assignment System

A commonly used method for evaluating the caliber of papers submitted to journals or scientific conferences is peer review. Conference organizers use conference management systems (CMS) to assign reviewers to submitted papers and invite suitable reviewers. Conventional CMS compute the paper assignment by using basic matching algorithms on the paper bids submitted by the reviewers. In contrast to

widely used CMSs, there is a Reviewer Assignment System (RAS) in this work, which has more functionalities. Initially, reviewer and submission profiles are automatically extracted by RAS as topic vectors. By using these profiles, reviewers can be automatically assigned to papers instead of going through the time-consuming and prone to error bidding procedure. [3]

2.3.4 Architecture of a Conference Management System Providing Advanced PaperAssignment Features

In order to facilitate the precise and accurate automatic assignment of reviewers to articles, this study offers the architecture and assignment management model of a conference management system. The system describes papers and reviewers' competencies using a taxonomy of keywords. The taxonomy's inferred hierarchical structure offers crucial further information about the semantic connections among the individual terms. It enables similarity metrics to determine how semantically similar two papers are, in addition to counting the number of keywords that perfectly match between them and the reviewer. Reviewers are free to express their explicit conflicts of interest (CoI) with articles and to bid on the papers they would like to review or not. [4]

2.3.5 A Comparative Review of Conference Management System

A succinct but thorough overview of Conference Management Systems (CMS) is given in this article, which also features popular platforms that conference organizers frequently utilize, like Microsoft CMT, COMS, EDAS, EasyChair, and ConfBay. With an emphasis on important areas including paper submission, the review procedure, registration, agenda and program management, virtual conference support, proceedings, and email correspondence, the advantages and characteristics of each platform are examined. By means of this comparative research, the many functionalities and unique contributions of each CMS are made apparent, providing conference organizers with useful insights to enable them to make informed selections that are specifically customized to their event requirements. [5]

2.3.6 An Efficient and Intelligent Conference Management System- Desired Services and Features

Journals and conferences provide a solid medium and platform for scientific endeavors. A conference's organization is a lengthy, challenging, and intricate process that involves numerous parties. Quality has become a highly sought-after topic due to the proliferation of conferences held worldwide and the volume of submissions to conferences and journals. There are many platforms and systems available now that offer different services and capabilities for planning conferences. To effectively manage the entire event, conference organizers must select one of the Conference Management Systems (CMS). This article compares thirty CMS and gives a list of desired features and services for an effective CMS. The following are the essential elements of an effective CMS: impartial evaluations; accuracy; dependability; speed; web-based/online-based; complete automation. [6]

2.3.7 Design Approach in Conference Management System with EZDESK Dashboard for Digital Ecosystem

The idea behind the conference management system is to handle and maintain pertinent files in several activities both before and after the conference. In actuality, the administrative procedures that are documented are part of the conference preparations. As a key component that drives overall implementation, the direction and modeling process in this instance should take the user experience (UX) into account using a simplified design approach to RMS (recognize, materialize, and scrutinize). This approach has been designed to assist in the development of prototypes with standards and usability aspects that align with user requirements. An application called EzDesk was created to help customers with application registration and document verification related to conference registration. A digital ecosystem, on the other hand, is a collection of linked IT resources that work together as a single entity. [7]

2.3.8 Automatic Topics Identification for Reviewer Assignment

The planning team would greatly benefit from having a Web-based management system that makes some of the numerous intricate and multifaceted tasks involved in scientific conference management a little bit simpler. Assigning submitted papers to appropriate reviewers involves the conference chair, reviewers, and authors. This is

one of these activities. When submitting a paper, authors typically need to fill out a form with the title of their work, an abstract, and a list of conference themes related to the topic of their submission. Reviewers must register and indicate, among other things, that they are knowledgeable about the conference subjects. Lastly, the conference chair must complete the review task while considering the data submitted by the reviewers and authors regarding their paper. [8]

2.3.9 ConfSys2: an improved web-based multi-conference management system

In this study, we present ConfSys2, an enhanced web-based system for managing multiple conferences as well as journal submission and publication. With additional capabilities to assist general chairs, program chairs, and program committees in managing academic conference procedures and in providing conference-related services to authors and conference participants, ConfSys2 is an advanced redesign of the ConfSys system. The insights gained from utilizing ConfSys have been integrated into Confsys2, which not only introduces new concepts like conference/journal series management, user-group-function management, and smart daemon in conference management to improve data sharing, reduce repetitive work, and make management work more flexible, but also implements a better user interface for Confsys's useful functions, like automatically/manually allocating paper to reviewers, debating, and rating paper. [9]

2.3.10 CyberChair: A Web-Based Groupware Application to Facilitate the Paper Reviewing Process

In this paper, CyberChair, an online groupware tool that facilitates the technical contribution evaluation process for conferences, is provided. The majority of the administrative duties related to the review process are handled by CyberChair, including author data storage, abstracts, camera-ready manuscripts, and reviews. Based on the reviews, it produces a number of summaries that help the Program Committee (PC) choose the best articles. Conflicting reviews are flagged by CyberChair, which also provides a way for reviewers to quickly resolve these issues through communication. O. Nierstrasz uses a pattern language to explain this review procedure in his work Identify the Champion. CyberChair is implemented using these patterns to support PCs. [10]

Chapter 3

System Requirements

3.1 Use Case Diagram

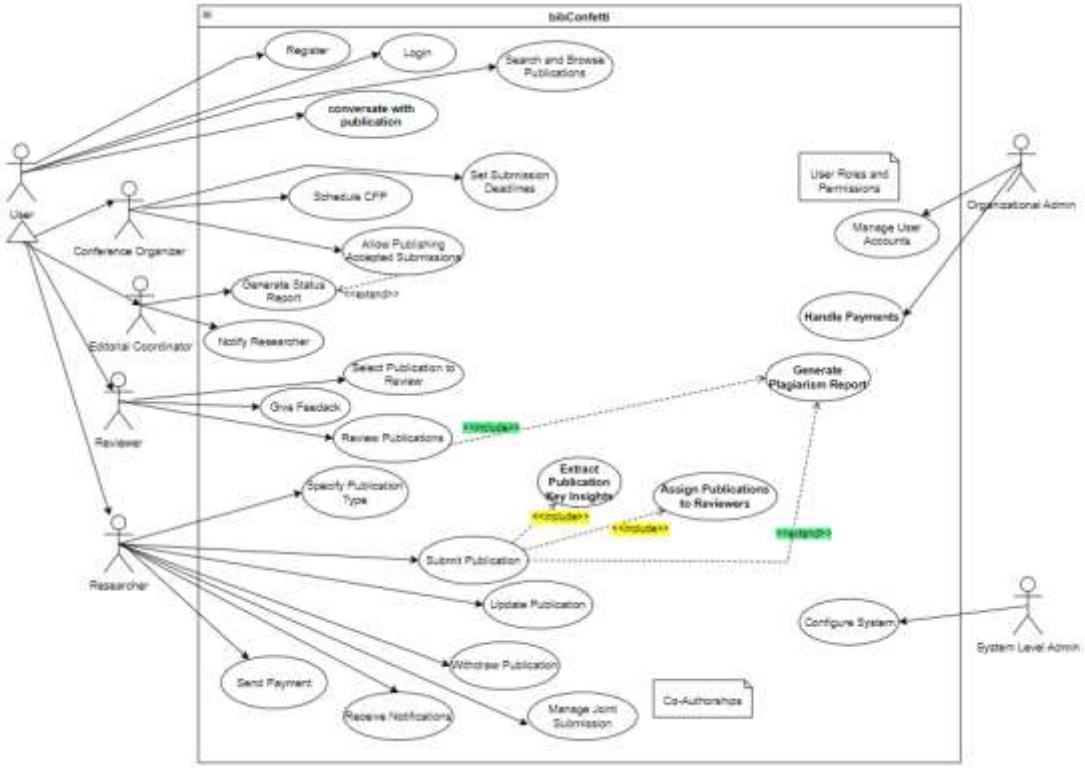

Figure 1: Use Case Diagram

Description The above use-case diagram shows the contact of the Users with the Web Application. The above diagram includes a user and an admin. The functionalities of the user are subdivided into more actors and their details are given below and how they interact with the system.

1. **Editorial Coordinator:**

Frequency of Use: Frequent user, particularly during the submission review and publication phases.

Product Functions: Access to editorial functionalities, including generating status reports, and notifying researchers.

Technical Expertise: Moderate technical proficiency, as the role involves coordinating the review process and managing publication-related tasks.

Security/Privilege Level: Moderate to high privilege level, with authority to oversee and facilitate the editorial workflow.

Educational Level/Experience: Typically educated with experience in academic publishing, editorial coordination, or related fields.

Duties: Produces updates on submitted research papers, interacts with scientists as well. Moreover, in coming up with briefs for publications the editorial coordinator has a significant position, and is responsible for overseeing the completion status of articles during review.

2. **Conference Organizer:**

Frequency of Use: Frequent user

Product Functions: Availability of the functionalities of conference management.

Technical Expertise: Moderate to high technical proficiency

Security/Privilege Level: High privilege level

Educational Level/Experience: Usually they have a background in event management or a related field and have coordinated academic conferences before.

Duties: Ensure that academic conferences are well planned, scheduled and managed in the system, which involves setting deadlines for submission, organizing conference timetables and having accepted papers published.

3. **Reviewing Committee:**

Frequency of Use: Variable, depending on the number of assigned reviews and ongoing publication submissions.

Product Functions: Access to review functionalities.

Technical Expertise: Moderate to high technical proficiency.

Security/Privilege Level: Moderate privilege level.

Educational Level/Experience: Typically holds an advanced degree in the relevant field and possesses expertise in the subject matter being reviewed.

Duties: Choose publications to review, give advice to authors, and detect plagiarism using an all-in-one tool. The Editorial Board aids in deciding whether a paper should be published or not depending on its academic value and novelty.

4. **Researcher:**

Frequency of Use: Variable, depending on the researcher's activity level, including submission and management of publications.

Product Functions: Access to a range of functionalities, such as specifying publication type, submitting and updating publications, managing joint submissions, accessing the plagiarism checker, and sending payments.

Technical Expertise: Moderate technical proficiency.

Security/Privilege Level: Moderate privilege level.

Educational Level/Experience: Varied.

Duties: The researcher submits publications, does publication details specification, makes joint submissions with co-authors and sends payments, while interacting with the plagiarism checker tool to ascertain the originality of the submitted work. The researcher should ensure that academic content submitted is accurate and timely.

5. **Organizational Admin:**

Frequency of Use: Intermittent

Product Functions: Access to administrative functionalities, including managing user accounts, defining roles and permissions, and handling payments.

Technical Expertise: Moderate to high technical proficiency

Security/Privilege Level: High privilege level

Educational Level/Experience: Typically well-educated with experience in system administration and organizational management.

Responsibilities: Manages user accounts, defines roles and permissions, and oversees payment-related processes. The organizational admin plays a crucial role in maintaining the integrity and security of the system, ensuring that user roles align with organizational needs, and overseeing financial transactions within the platform.

6. **System Level Admin:**

Frequency of Use: Infrequent

Product Functions: Access to advanced system configuration settings.

Technical Expertise: High technical proficiency.

Security/Privilege Level: Highest privilege level

Educational Level/Experience: Typically possesses advanced education and extensive experience in system administration, software architecture, and security protocols.

Duties: maintain and configure the software

3.2 Functional Requirements:

3.2.1 Conference Organizer

3.2.1.1 Schedule Conferences

Table 1: Schedule Conferences

Requirement ID:	FR01	
Requirement:	Schedule Call for Paper/Publication	
Actor(s):	Conference Organizer	
Pre-Conditions:	Conference details (date, time, venue) are not scheduled.	
Priority:	Medium	
Basic Flow:	The conference organizer initiates the scheduling process, providing necessary details such as date, time, and venue through the system interface. Upon confirmation, the system records and saves the conference schedule.	
Actor Actions		**System Response**
1	Conference organizer accesses the scheduling feature.	The system validates the provided information.
2	Inputs conference details, including date, time, and venue.	If valid, it confirms successful scheduling and updates the conference schedule.
3	Submits the scheduling request through the system.	Sends a notification to the organizer about the successful scheduling.
Alternative Course of Action (if any)		
Actor Action		**System Response**
	If there is a scheduling conflict with a previously scheduled conference, the organizer adjusts the timing or venue.	The system checks for conflicts and updates the schedule, providing confirmation upon successful adjustment

		or alerts if conflicts persist.

3.2.1.2 Set Submission Deadlines

Table 2: Set Submission Deadlines

Requirement ID:	FR02	
Requirement:	Set Submission Deadlines	
Actor(s):	Conference Organizer	
Pre-Conditions:	Conference details are available.	
Priority:	High	
Basic Flow:	Accessing the conference management system, navigating to the submission deadline settings, and defining the deadlines for authors to submit their publications.	
Actor Actions		**System Response**
1	Conference Organizer logs into the conference management system.	The system verifies the organizer's credentials and authenticates the login.
2	The organizer navigates to the submission deadline configuration section.	The system presents the submission deadline configuration interface.
3	The organizer specifies the submission deadlines for authors.	The system saves the specified deadlines in the database and confirms successful configuration.
Alternative Course of Action (if any)		
Actor Action		**System Response**
	The organizer encounters an issue in setting submission deadlines (e.g., technical error or invalid input)	The system will display an error message and prompt the organizer to correct the issue before saving the configuration.

3.2.1.3 Publish Accepted Submissions

Table 3: Publish Accepted Submissions

Requirement ID:	FR03
Requirement:	Allow Publishing Accepted Submissions
Actor(s):	Conference Organizer
Pre-Conditions:	Accepted submissions have been finalized and payments have

	been done.	
Priority:	High	
Basic Flow:	Conference Organizer selects the accepted submissions for publication. System processes the selected submissions for publishing.	
Actor Actions		**System Response**
1	Conference Organizer selects accepted submissions.	System processes the selection.
2	Conference Organizer confirms the publishing.	System publishes the submissions.
Alternative Course of Action (if any)		
Actor Action		**System Response**
	Conference Organizer contacts technical support. (If there are technical issues during the publishing process)	System provides assistance and resolves the issue.

3.2.2 Editorial Coordinator

3.2.2.1 Notify Researcher

Table 4: Notify Researcher

Requirement ID:	FR04	
Requirement:	Notify Researcher	
Actor(s):	Editorial Coordinator	
Pre-Conditions:	There is a change in status of the publication.	
Priority:	High	
Basic Flow:	Editorial Coordinator selects the publication and composes a notification. Editorial Coordinator sends the notification to the specified researcher.	
Actor Actions		**System Response**
1	Editorial Coordinator selects	System displays a list of publications for

	publication.	selection.
2	Editorial Coordinator composes notification (It could include notifying researchers about the assignment of reviewers, providing feedback, or requesting revisions.)	System provides a text editor for composing the notification.
3	Editorial Coordinator sends notification	System sends the notification to the specified researcher.
Alternative Course of Action (if any)		
Actor Action		**System Response**
	Editorial Coordinator encounters an issue while sending the notification (e.g., email failure)	1a. System displays an error message. 1b. System logs the issue for further investigation.

3.2.2.2 Generate Status Report

Table 5: Generate Status Report

Requirement ID:	FR05	
Requirement:	Generate Status Report	
Actor(s):	Editorial Coordinator	
Pre-Conditions:	Publications assigned to reviewers.	
Priority:	Medium	
Basic Flow:	The Editorial Coordinator selects the option to generate a status report for the assigned publications. The system collects and compiles the status information of each assigned publication.	
Actor Actions		**System Response**
1	Editorial Coordinator selects "Generate Status Report" option.	System collects and compiles status information.
Alternative Course of Action (if any)		
Actor Action		**System Response**
	Editorial Coordinator cancels the operation.	System cancels the report generation.

3.2.3 Reviewer

3.2.3.1 Select Publication to Review

Table 6: Select Publication to Review

Requirement ID:	FR06	
Requirement:	Select Publication to Review	
Actor(s):	Reviewer	
Pre-Conditions:	The system has assigned publications for review using AI-NLP.	
Priority:	High	
Basic Flow:	The Reviewer, after logging in, selects a publication from the list of assigned publications.The chosen publication is marked for review.	
Actor Actions		**System Response**
1	Reviewer selects a publication	System assigns the publication.
Alternative Course of Action (if any)		
Actor Action		**System Response**
	Reviewer requests more information	System provides additional details.

3.2.3.2 Review Publications

Table 7: Review Publications

Requirement ID:	FR07	
Requirement:	Review Publications	
Actor(s):	Reviewer	
Pre-Conditions:	Reviewer has logged in and selected a publication for review.	
Priority:	High Reviewer selects a publication for review. System generates a plagiarism report for the selected publication. Reviewer provides feedback on the publication. System records the review and feedback.	
Basic Flow:		
Actor Actions		**System Response**

1	Reviewer selects a publication for review.	System generates a plagiarism report for the selected publication.
2	Reviewer provides feedback on the publication.	System records the review and feedback.

3.2.3.3 Give Feedback

Table 8: Give Feedback

Requirement ID:	FR08	
Requirement:	Give Feedback	
Actor(s):	Reviewer	
Pre-Conditions:	Reviewer has reviewed a publication.	
Priority:	High	
Basic Flow:	Reviewer accesses the feedback section for the reviewed publication. Reviewer provides detailed feedback on the strengths and weaknesses of the publication. System records the feedback.	
Actor Actions		**System Response**
1	Reviewer accesses the feedback section for the reviewed publication.	System displays the feedback form for the selected publication.
2	Reviewer provides detailed feedback on the strengths and weaknesses of the publication.	System records the feedback.

3.2.4 Researcher

3.2.4.1 Specify Publication Type

Table 9: Specify Publication Type

Requirement ID:	FR9
Requirement:	Specify Publication Type

Actor(s):	Researcher	
Pre-Conditions:	Researcher is logged into the system.	
Priority:	Low	
Basic Flow:	Researcher navigates to the submission interface. Researcher selects the option to specify the type of publication (e.g., paper, journal, article). Researcher submits the specified publication type. System records the specified publication type.	
Actor Actions		**System Response**
1	Researcher navigates to the submission interface.	System displays the submission interface.
2	Researcher selects the option to specify the type of publication.	System presents a list of publication types (paper, journal, article).
3	Researcher submits the specified publication type.	System records the specified publication type.

3.2.4.2 Submit Publication

Table 10: Submit Publication

Requirement ID:	FR10	
Requirement:	Submit Publication	
Actor(s):	Researcher	
Pre-Conditions:	Researcher is logged into the system and has specified the publication type.	
Priority:	High	
Basic Flow:	1. Researcher initiates the submission process for the publication. 2. System validates the submission details, extracts key insights using AI. 3. Researcher confirms the submission. 4. System records the submitted publication, auto-assigns reviewers using AI, and displays a confirmation message.	
Actor Actions		**System Response**
1	Researcher initiates the submission	System displays the submission form and fill abstract, keywords on its own.

	process for the publication.	
2	Researcher confirms the submission.	2a. System displays a confirmation message. 2b. System assign reviewers. 2c. System records the submitted publication.

3.2.4.3 Update Publication

Table 11: Update Publication

Requirement ID:	FR11	
Requirement:	Update Publication	
Actor(s):	Researcher	
Pre-Conditions:	Researcher is logged into the system, and a publication has been reviewed with suggested changes.	
Priority:	High	
Basic Flow:	Researcher selects the option to upload an updated version of the publication after receiving feedback from the reviewer. System prompts the Researcher to upload the corrected version of the publication. Researcher uploads the updated publication file with reviewer-suggested changes. Researcher confirms the submission. System records the updated publication information.	
Actor Actions		**System Response**
1	Researcher selects the option to upload an updated version of the publication.	System prompts the Researcher to upload the corrected version of the publication.
2	Researcher uploads the updated publication file with reviewer-suggested changes.	System acknowledges the successful upload.
3	Researcher confirms the submission.	System records the updated publication information.

3.2.4.4 Withdraw Publication

Table 12: Withdraw Publication

Requirement ID:	FR12
Requirement:	Withdraw Publication
Actor(s):	Researcher

Pre-Conditions:	Researcher is logged into the system, and the publication to be withdrawn exists in the system.
Priority:	High
Basic Flow:	Researcher selects the option to withdraw a publication. System displays a list of the Researcher's publications. Researcher chooses the publication to withdraw. System prompts the Researcher to confirm the withdrawal. Researcher confirms the withdrawal. System removes the publication from public access and updates the status.

Actor Actions		**System Response**
1	Researcher selects the option to withdraw a publication.	System displays a list of the Researcher's publications.
2	Researcher chooses the publication to withdraw.	System prompts the Researcher to confirm the withdrawal.
3	Researcher confirms the withdrawal.	System removes the publication from public access and updates the status.

3.2.4.5 Receive Notifications

Table 13: Receive Notifications

Requirement ID:	FR13
Requirement:	Receive Notifications
Actor(s):	Researcher
Pre-Conditions:	Researcher is logged into the system, and there is a change in the status of the submitted publication.
Priority:	Medium
Basic Flow:	Researcher logs into the system. System checks for any updates or changes in the status of the submitted publication. If there is a change, the system sends a notification to the Researcher. Researcher views the notification to learn about the status change.

Actor Actions		**System Response**
1	Researcher logs into the system.	1a. System checks for any updates or changes in the status of the submitted publication.

		1b. If there is a change, the system sends a notification to the Researcher.
2	Researcher views the notification to learn about the status change.	

3.2.4.6 Manage Joint Submission (Co-Authorships)

Table 14: Manage Joint Submission (Co-Authorships)

Requirement ID:	FR14	
Requirement:	Manage Joint Submission (Co-Authorships)	
Actor(s):	Researcher	
Pre-Conditions:	Researcher is logged into the system, and the publication is a joint submission with multiple co-authors.	
Priority:	High	
Basic Flow:	System displays a list of joint submissions associated with the Researcher. Researcher selects a specific joint submission for management. System allows the Researcher to add or remove co-authors, update information, and manage collaboration settings. Researcher confirms the changes and updates to the joint submission.	
Actor Actions		**System Response**
1	Researcher selects the option to manage joint submissions in the user dashboard.	System displays a list of joint submissions associated with the Researcher.
2	Researcher selects a specific joint submission for management. Researcher confirms the changes and updates to the joint submission.	System allows the Researcher to add or remove co-authors, update information, and manage collaboration settings.

3.2.4.7 Send Payment

Table 15: Send Payment

Requirement ID:	FR15
Requirement:	Send Payment
Actor(s):	Researcher

Pre-Conditions:	Researcher is logged into the system, and the publication has been accepted for publication.	
Priority:	High	
Basic Flow:	Researcher navigates to the payment section for the accepted publication. System displays the payment details, including the amount to be paid. Researcher selects the preferred payment method, provides necessary payment information, and completes the payment. System processes the payment and sends a payment confirmation and receipt to the Researcher.	
Actor Actions		**System Response**
1	Researcher navigates to the payment section for the accepted publication. Researcher selects the preferred payment method.	System displays the payment details, including the amount to be paid.
2	Researcher provides necessary payment information.	System processes the payment through the payment processing module and sends a payment confirmation and receipt to the Researcher.
Alternative Course of Action (if any)		
Actor Action		**System Response**
1	Researcher receives an error message indicating the issue If the payment processing module encounters an error during the payment transaction	Researcher can choose an alternative payment method or contact customer support for assistance.
2	Researcher cancels the payment transaction If the Researcher decides to cancel the payment.	System cancels the payment process and notifies the Researcher that the payment has been canceled.

3.2.4.8 Generate Plagiarism Report

Table 16: Generate Plagiarism Report

Requirement ID:	FR16
Requirement:	Generate Plagiarism Report

Actor(s):	Researcher	
Pre-Conditions:	Researcher has submitted a publication and wants to check it for plagiarism.	
Priority:	High	
Basic Flow:	The Researcher confirms the submission, gains access to the Plagiarism Checker, and uploads the publication. The system processes the document, generates a Plagiarism Report, and notifies the Researcher of the results.	
Actor Actions		**System Response**
1	Researcher confirms the submission.	System acknowledges the submission and grants access to additional features.
2	Researcher accesses the Plagiarism Checker.	System presents the Plagiarism Checker interface.
3	Researcher uploads the publication.	System confirms the successful upload and initiates the plagiarism check.
4	Researcher sees a processing indicator.	System processes the document.
5	Researcher receives a notification that the report is ready for review.	System generates a Plagiarism Report.
6	Researcher receives the Plagiarism Report.	System displays the Plagiarism Report with details on any detected plagiarism.
Alternative Course of Action (if any)		
Actor Action		**System Response**
1	Researcher may retry the plagiarism check or contact support for assistance.	Researcher receives an error message if the Plagiarism Checker encounters technical issues during processing.

3.2.5 User

3.2.5.1 Register

Table 17: Register

Requirement ID:	FR17

Requirement:	Register	
Actor(s):	User	
Pre-Conditions:	None	
Priority:	High	
Basic Flow:	User accesses the system and chooses the registration option. User provides required registration information, including username, email, and password. System validates the information and registers the user.	
Actor Actions		**System Response**
1	User initiates the registration process	System displays the registration form.
2	User submits the registration form	System verifies the information.
3	User clicks the confirmation link	System confirms the registration and authenticates the user.
Alternative Course of Action (if any)		
Actor Action		**System Response**
1	user does not confirm the registration within a specified time	The system sends a reminder email.
2	User provided information is incomplete or invalid	The system displays an error message, and the user is prompted to correct the information.

3.2.5.2 Login

Table 18: Login

Requirement ID:	FR18	
Requirement:	Login	
Actor(s):	User	
Pre-Conditions:	Registered user account	
Priority:	High	
Basic Flow:	User provides login credentials (email and password). System verifies the credentials. If credentials are valid, the system logs in the user.	
Actor Actions		**System Response**

1	User provides login credentials	System verifies the credentials.
2	Provided credentials are valid	System logs in the user.
Alternative Course of Action (if any)		
Actor Action		**System Response**
1	User provided credentials are invalid	System displays an error message, and the user is prompted to retry.

3.2.5.3 Search and Browse Publications

Table 19: Search and Browse Publications

Requirement ID:	FR19	
Requirement:	Search and Browse Publications	
Actor(s):	User	
Pre-Conditions:	User is logged in	
Priority:	High	
Basic Flow:	User logs in and navigates to the search and browse publications option. User enters search criteria or browses available publications. System displays relevant publications based on the search or browsing.	
Actor Actions		**System Response**
1	User logs in and selects search and browse publications	System displays the publication interface.
2	User enters search criteria or browses	System presents relevant publications.
Alternative Course of Action (if any)		
Actor Action		**System Response**
1		If there are no publications matching the criteria, the system informs the user and suggests modifying the search parameters.

3.2.5.4 Conversate with publication

Table 20: Conversate with publication

Requirement ID:	FR20

Requirement:	Conversate with publication	
Actor(s):	User (researcher, reviewer, editorial coordinator)	
Pre-Conditions:	User is logged into the system, and the publication submission process is complete.	
Priority:	Medium to High, depending on the project requirements.	
Basic Flow:	User initiates a conversation related to a submitted publication. System provides a real-time messaging interface within the publication. Users can post messages, ask questions, or provide clarifications.	
Actor Actions		**System Response**
1	User initiates a conversation related to a submitted publication.	System provides a messaging interface.
2	Users post messages, ask questions, or provide clarifications.	System enables real-time messaging and displays user messages.

3.2.6 System Level Admin

3.2.6.1 Configure System

Table 21: Configure System

Requirement ID:	FR21	
Requirement:	Configure System	
Actor(s):	System Level Admin	
Pre-Conditions:	System Level Admin is authenticated	
Priority:	High	
Basic Flow:	System Level Admin logs in and navigates to the system configuration section. System Level Admin configures various system settings, including user roles, permissions, and other system parameters.	
Actor Actions		**System Response**
1	System Level Admin logs in and selects system configuration	System displays the configuration interface.
2	System Level Admin configures system settings	System updates the configuration and provides a confirmation message.

Alternative Course of Action (if any)		
Actor Action		**System Response**
1		If there are errors in the configuration, the system prompts the admin to correct them and ensures data integrity before applying changes.

3.2.7 Organizational Admin

3.2.7.1 Manage User Accounts (User Roles and Permissions)

Table 22: Manage User Accounts

Requirement ID:	FR22	
Requirement:	Manage User Accounts (User Roles and Permissions)	
Actor(s):	Organizational Admin	
Pre-Conditions:	Organizational Admin is authenticated	
Priority:	High	
Basic Flow:	Organizational Admin logs in and accesses the user management section. Organizational Admin manages user accounts by assigning roles and permissions, creating new accounts, or modifying existing ones.	
Actor Actions		**System Response**
1	Organizational Admin logs in and navigates to user management	System displays the user accounts interface.
2	Organizational Admin assigns roles and permissions to users	System updates user accounts and provides a confirmation message.
Alternative Course of Action (if any)		
Actor Action		**System Response**
1		If there are errors in the configuration, the system prompts the admin to correct them and ensures data integrity before applying changes.

3.2.7.2 Handle Payment

Table 23: Handle Payment

Requirement ID:	FR23	
Requirement:	Handle Payments	
Actor(s):	Organizational Admin	
Pre-Conditions:	Organizational Admin is authenticated, payment processing module is operational	
Priority:	High	
Basic Flow:	Organizational Admin logs in and accesses the payment management section. Organizational Admin reviews and processes payments, ensuring all financial transactions are accurate and complete.	
Actor Actions		**System Response**
1	Organizational Admin logs in and navigates to the payment management	System displays the payment interface.
2	Organizational Admin reviews and processes payments	System updates payment status and transaction records, providing confirmation.
Alternative Course of Action (if any)		
Actor Action		**System Response**
1		If there are discrepancies in payment records, the system flags them for manual review by the admin, ensuring financial accuracy and integrity.

3.3 Interface Requirements

3.3.1 User Interface:

The user interfaces for the Academic Conference & Research Article Management System will be designed to provide an intuitive and efficient user experience for various user classes. Sample screen images and layouts will be presented in the user interface specification document, ensuring consistency and adherence to GUI standards. Standard buttons such as "Submit," "Update," and "Withdraw" will be uniformly implemented across interfaces. Keyboard shortcuts will be incorporated where applicable. Error messages will follow a standardized format for clarity, and a comprehensive help function will be available.

3.3.2 Hardware Interface:

The software is designed to operate on standard computing hardware, including desktops, laptops, and servers. It does not impose specific hardware requirements beyond those expected for modern web applications. The system will interact with hardware components for data storage and retrieval. Hardware interfaces are standard and do not necessitate specialized protocols.

3.3.3 Software Interface:

The system interacts with a NOSQL database of MongoDB for data storage and retrieval. The user interfaces are accessed through standard web browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari. Integration with a third-party plagiarism checking service for validating the authenticity of submitted content.

It also includes interaction with a payment processing service for handling financial transactions. The web application is developed using specific server-side technologies of Node.js.

Data sharing mechanisms are standard, utilizing the database for persistent storage and retrieval and data items shared across software components include user credentials, conference details, and publication information.

3.4 Database Requirements

The system interacts with a NOSQL database of MongoDB for data storage and retrieval. The user interfaces are accessed through standard web browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari. Integration with a third-party plagiarism checking service for validating the authenticity of submitted content.

3.5 Non-Functional Requirements:

3.5.1 Performance Requirements:

- The system must provide fast response times for user interactions.
- Auto assignment, extracting key insights, uploading submission should be completed within seconds.

3.5.2 Safety Requirements:

- The software must ensure data security and user privacy.
- User authentication is required to access certain features.

3.5.3 Security Requirements:

- The system should have measures in place to prevent unauthorized access.
- Sensitive user data must be encrypted and protected.

3.5.4 Quality Attributes:

- The system should be user-friendly and easy to navigate.
- It must be highly reliable and available 24/7 for user access.

Business rules:

Users, regardless of their roles (Conference Organizer, Editorial Coordinator, Reviewing Committee, Researcher, User, System Level Admin, and Organizational Admin), must authenticate themselves through a secure login process before accessing the application.

3.6 Project Feasibility

This project is an online platform where a user has access to application where a user can upload publication and arrange conferences accordingly.

3.6.1 Technical Feasibility

This system is a web application. We developed the system so that it can be scalable, separate implementations of logic, and easy to maintain and upgrade. This system was

developed by keeping in mind the requirements, risks, and security issues related to the system.

3.6.2 Operational Feasibility

All the basic functionalities which are mentioned in requirements are fulfilled and system is operating smoothly.

3.6.3 Legal & Ethical Feasibility

Security to data of users and authenticated users are allowed to login through proper authentication and system fulfils its legal and ethical aspects.

3.7 Analysis Models

This is workflow model showing user hierarchal flow that how our actors are interacting with one another.

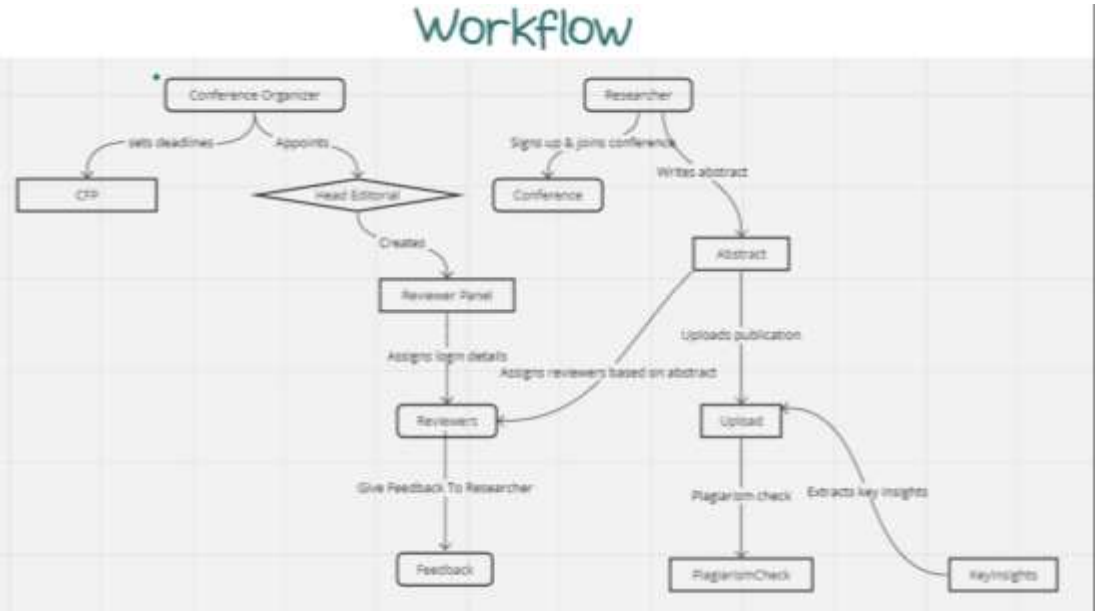

Figure 2: Workflow hierarchal order

Description:

The above workflow model showing user hierarchal flow that how our actors are interacting with one another.

3.8 Conclusion

In this chapter, we discussed requirements and functionalities performed by the system and what are roles of different actors and how they are interconnected with one another in the system.

Chapter 4
System Design

In this chapter, we have shown the design approach, system architecture, and user interfaces of our web application. We have shown the complete user interfaces of registration, admin module, user module etc.

4.1 Design Approach

The system will follow three-tier architecture which will include presentation layer, server/application layer and database layer.

4.2 Architecture:

The system follows a three-tier architecture, encompassing the following tiers:

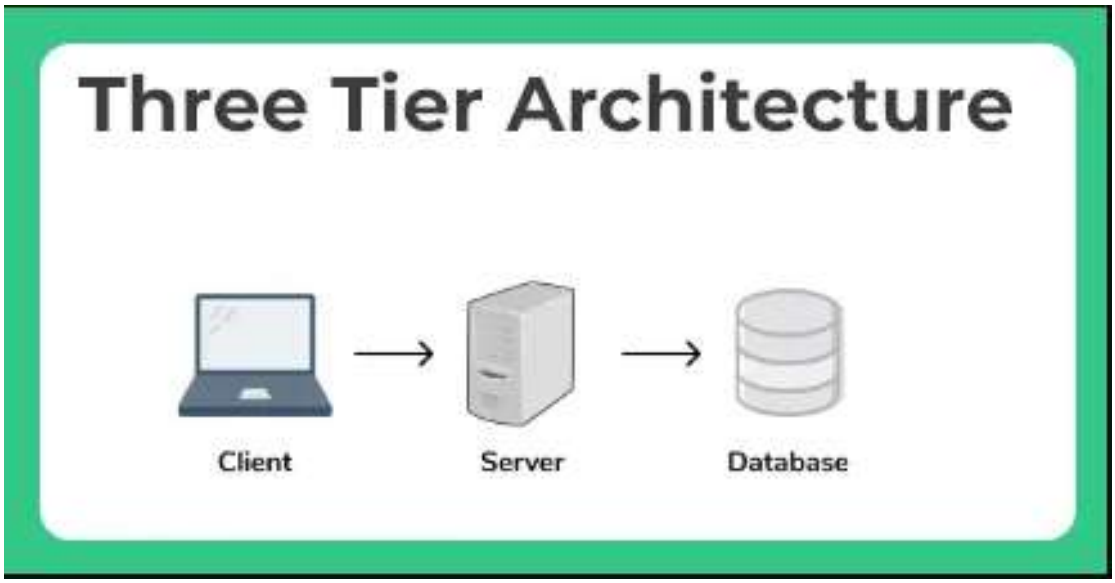

Figure 3: Architecture

Presentation Tier:

Represents the user interface layer where users interact with the system. It utilizes React/Next.js for creating an interactive and responsive web interface.

Application (Logic) Tier:

Houses the business logic and manages communication between the presentation and data tiers. Implemented using Node.js, python and Express.js, this tier processes requests, orchestrates the system's functionality, and interfaces with the data tier.

Data Tier:

Manages data storage and retrieval, interacting with MongoDB as the database system. This tier ensures efficient data management for the entire system.

4.3 Logical Design

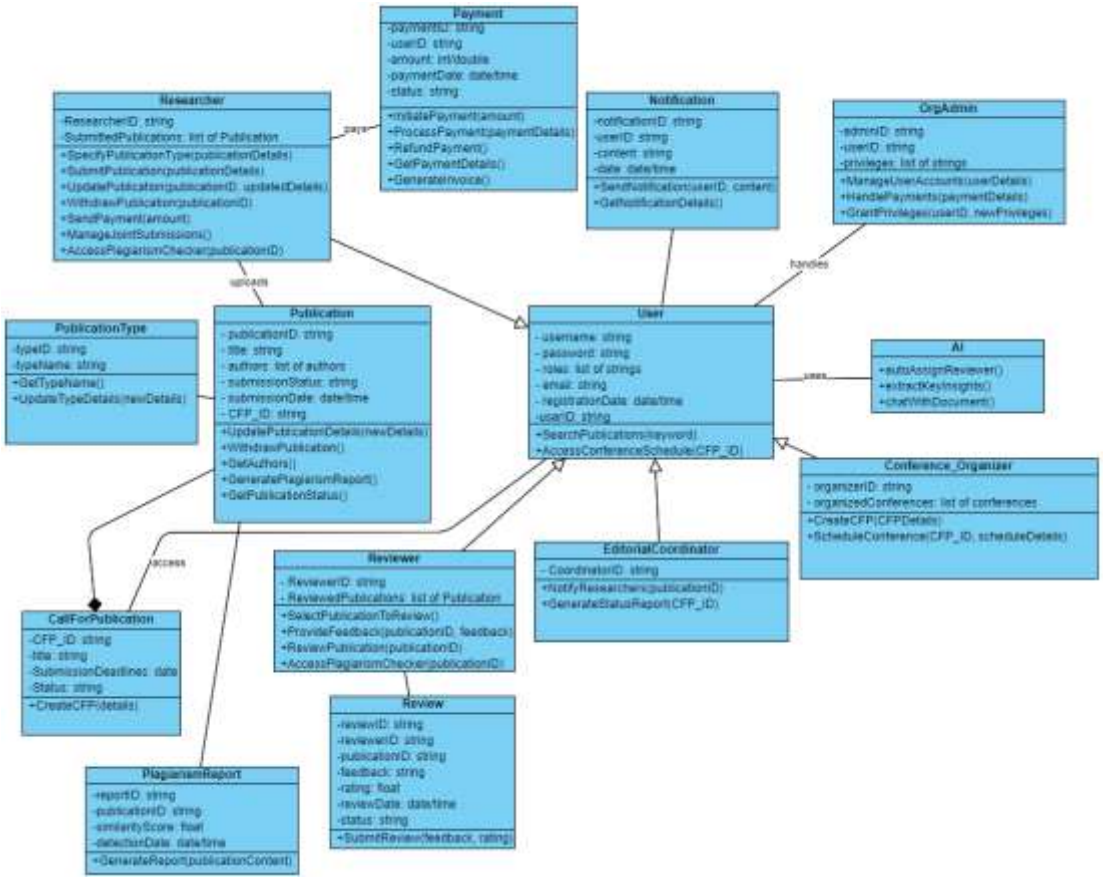

Figure 4: Class Diagram

Description: In this diagram, different classes which are used in our system are shown and how are they interacting with system what are different attributes of different actors are shown.

4.4 Dynamic View

4.4.1 Activity Diagrams

4.4.1.1 Researcher

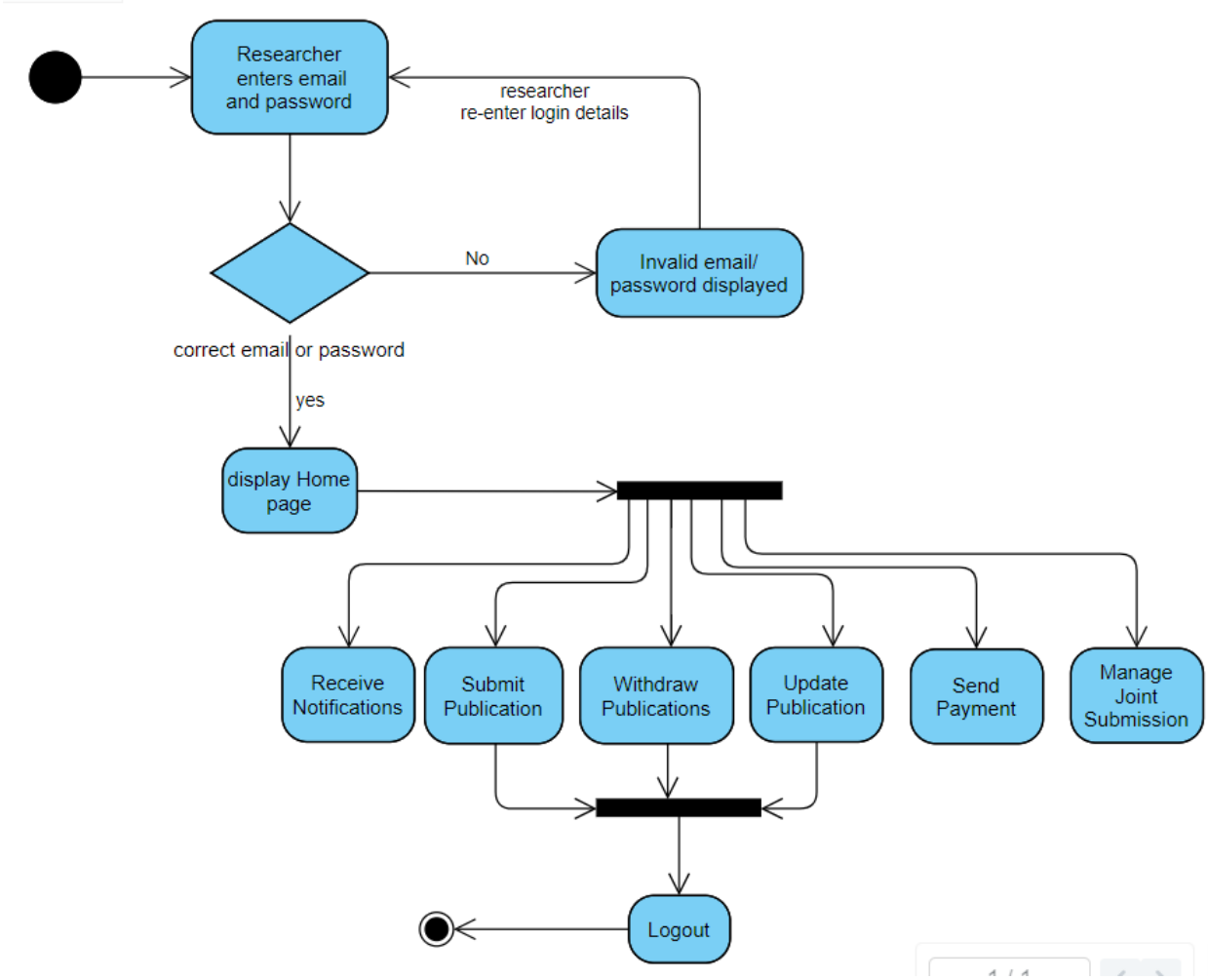

Figure 5: Activity Diagram for Researcher

Description:

In this above activity diagram researchers activities are shown and flow of how researcher performs his activities is described.

4.4.1.2 Reviewing Committee

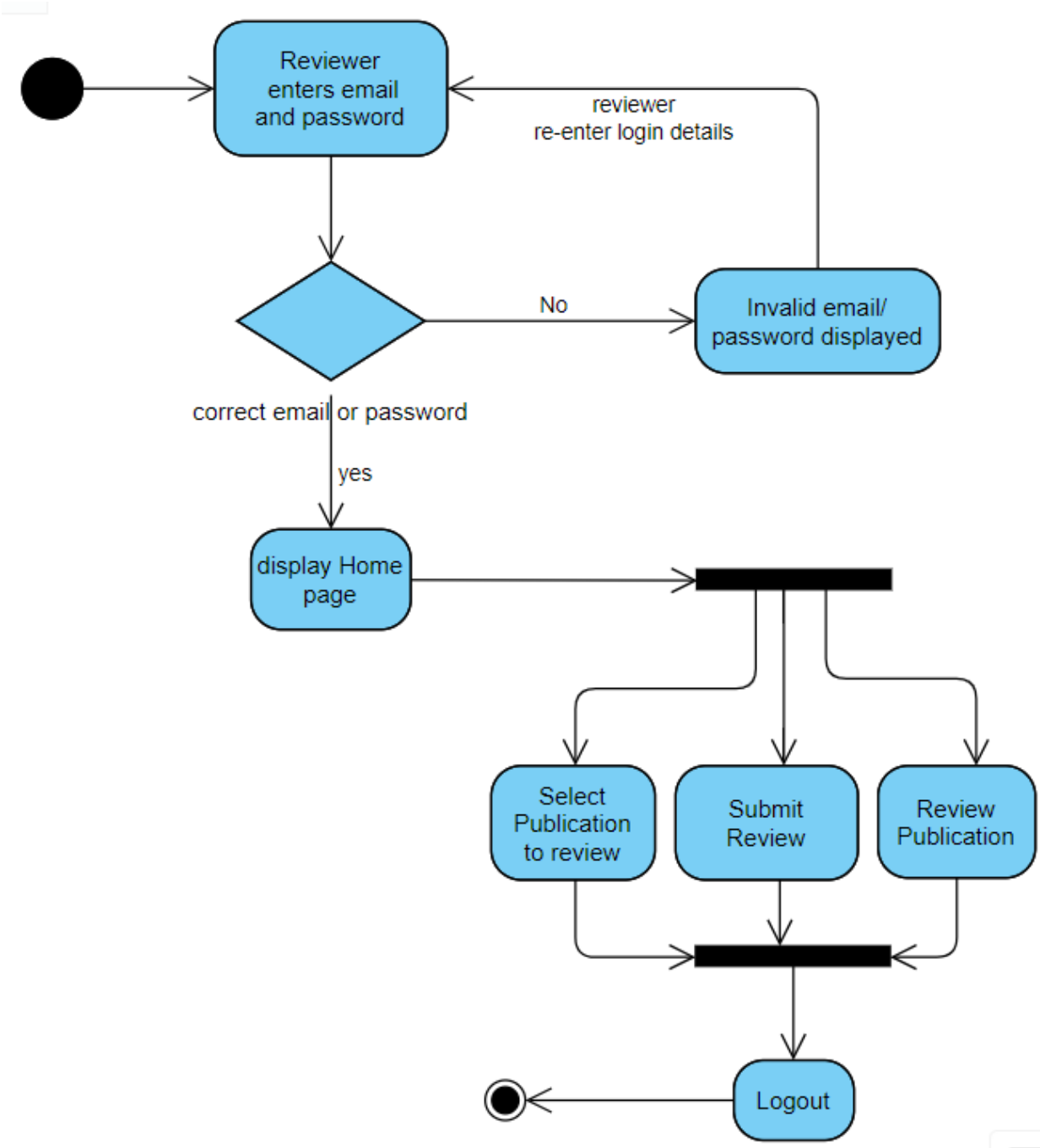

Figure 6: Activity Diagram for Reviewing Committee

Description:

In this activity diagram, reviewing committee activity is shown how they interact with the system and performs their functionalities.

4.4.1.3 Editorial Coordinator

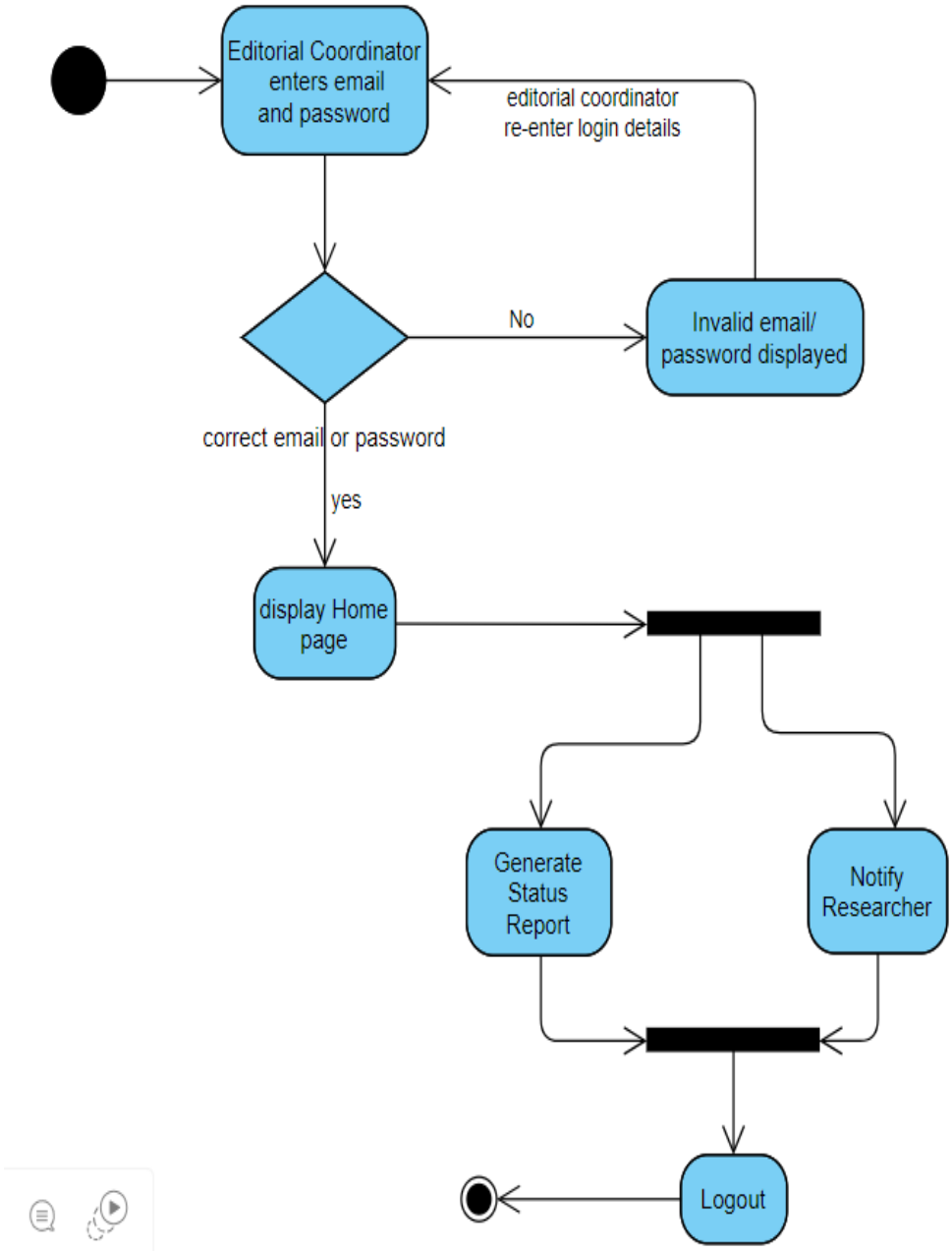

Figure 7: Activity Diagram for Editorial Coordinator

Description:

In this activity diagram, editorial coordinator activity is shown how they interact with the system and performs their functionalities.

4.4.1.4 Conference Organizer

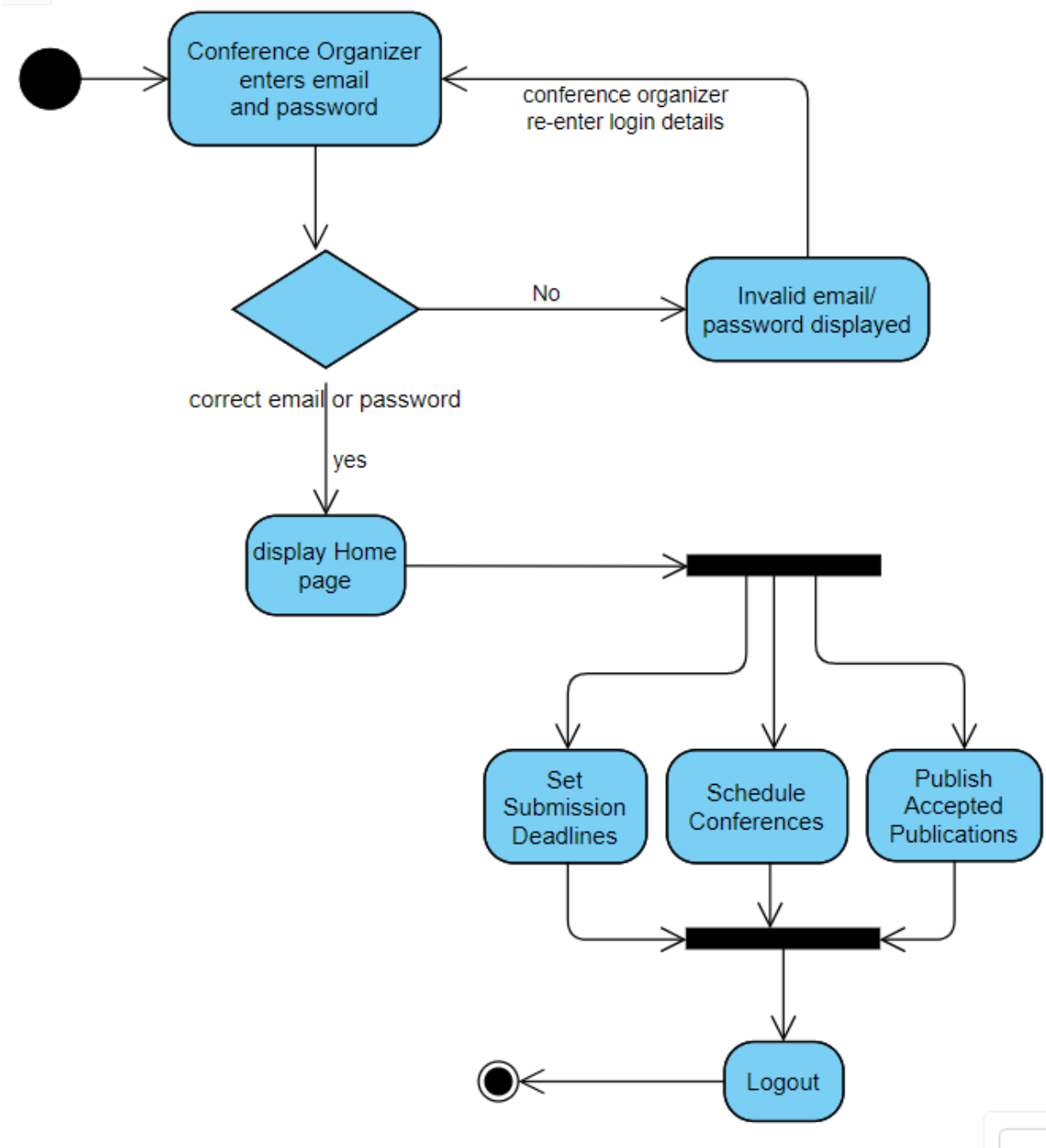

Figure 8: Activity Diagram for Conference Organizer

Description:

In this activity diagram, conference organizer activity is shown how they interact with the system and performs its activities of arranging conferences and scheduling conferences.

4.4.1.5 Organizational Admin

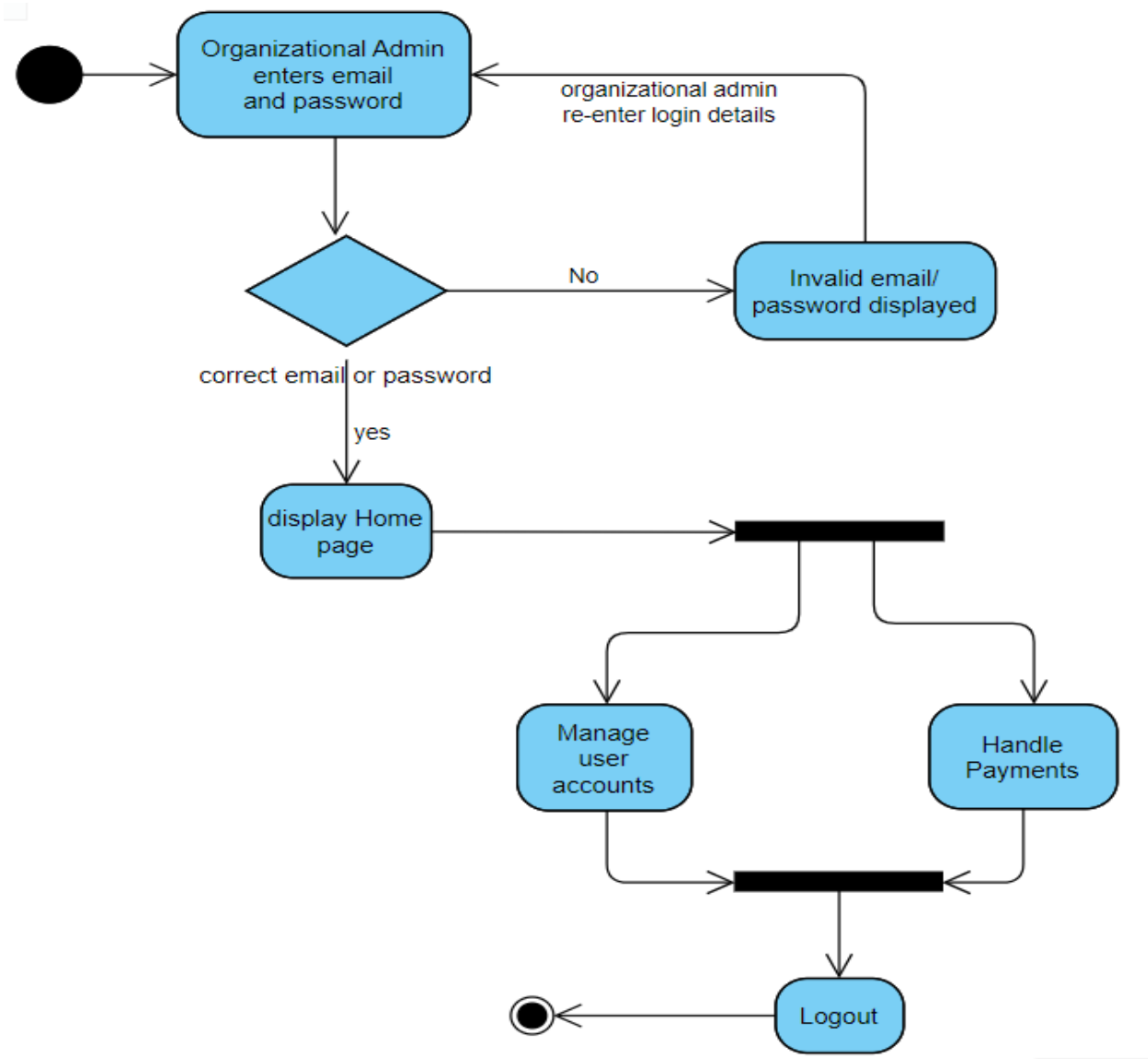

Figure 9: Activity Diagram for Admin

Description:

In this activity diagram, admin activities are shown and how admin interact with system and responsible for managing these critical activities are shown in this figure.

4.4.2 Sequence Diagrams

4.4.2.1 User

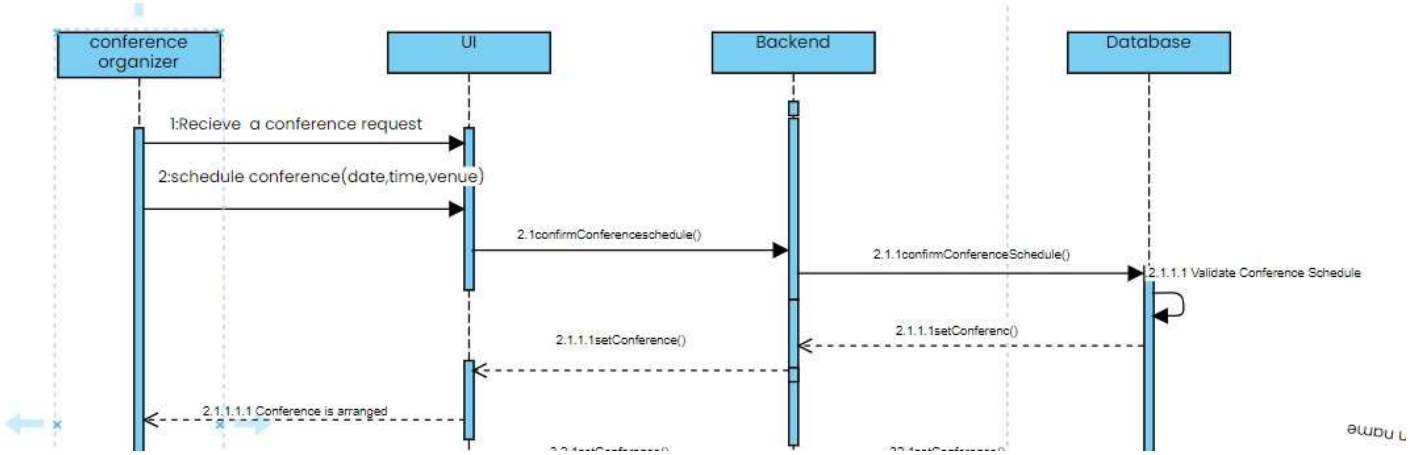

Figure 10: Sequence Diagram for Scheduling Conference

Description:

In this diagram, sequence of how conference organizer organizes a conference so user can join these conferences.

4.4.2.2 Conference Organizer

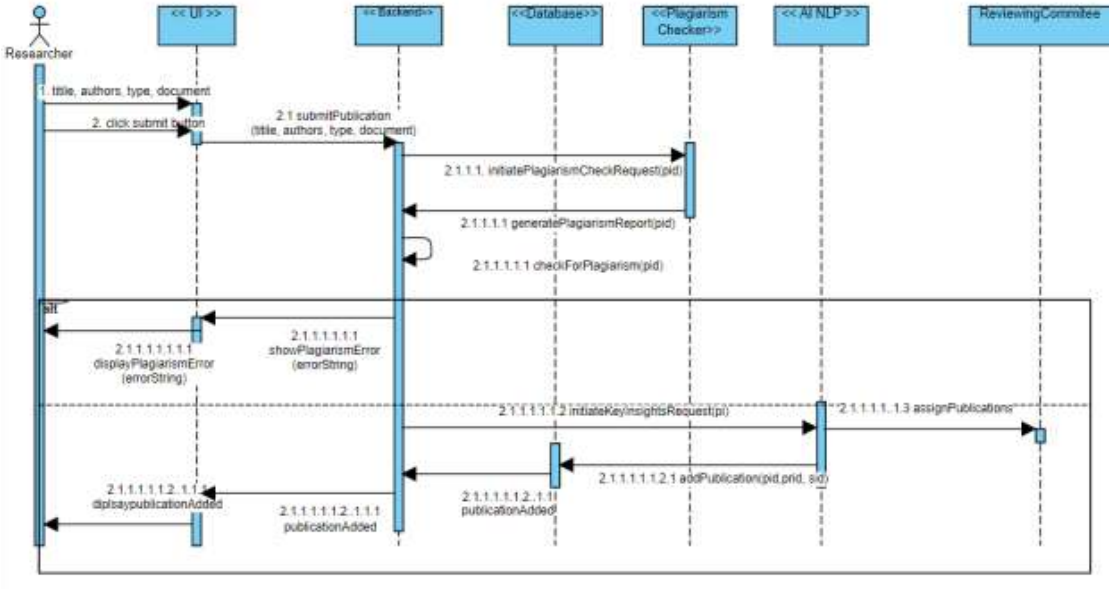

Figure 11: Sequence Diagram for submission Deadline.

Description:

In this Diagram, conference organizer is setting submission deadline for authors for submitting their publications.

4.4.2.3 Researcher

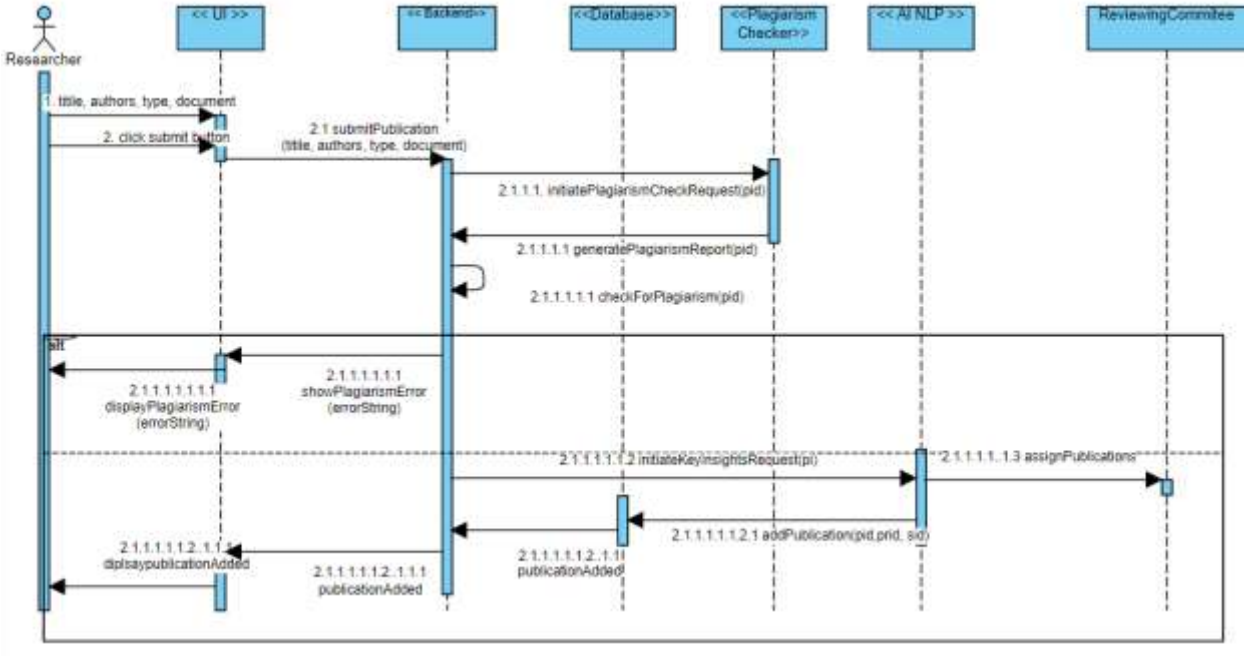

Figure 12: Sequence Diagram for Submit Publication

Description:

In this diagram, researcher submits a publication and how a publication of a researcher is uploaded is described in this diagram.

4.5 Deployment Design

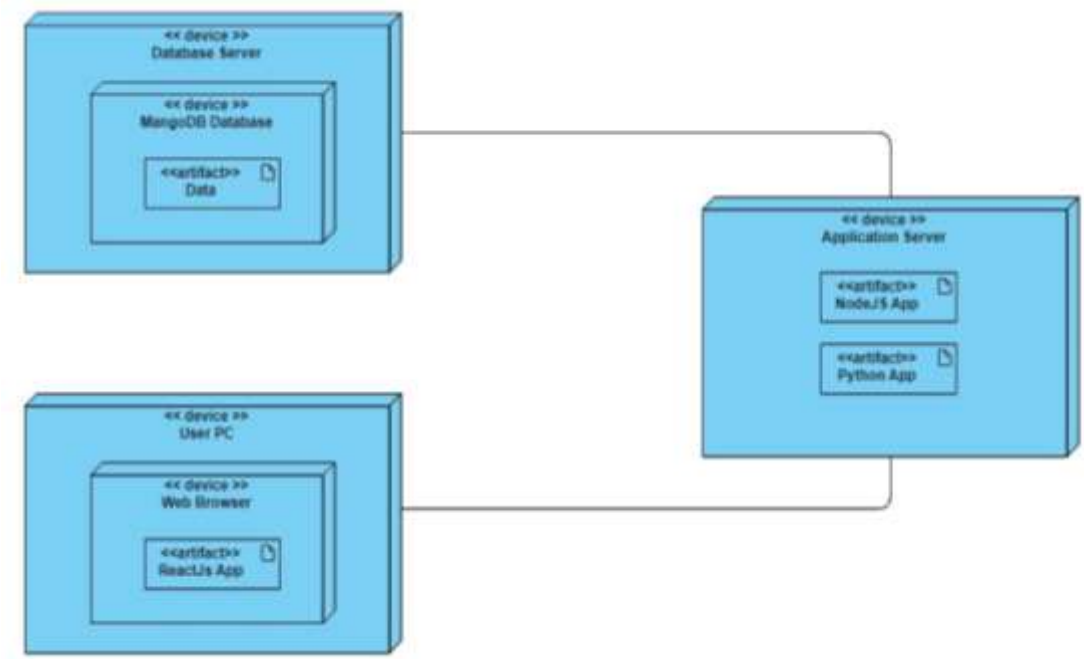

Figure 13: Deployment Diagram

Description:

In this diagram, the overall application technologies are shown and how they can be deployed on servers.

4.6 Data Models

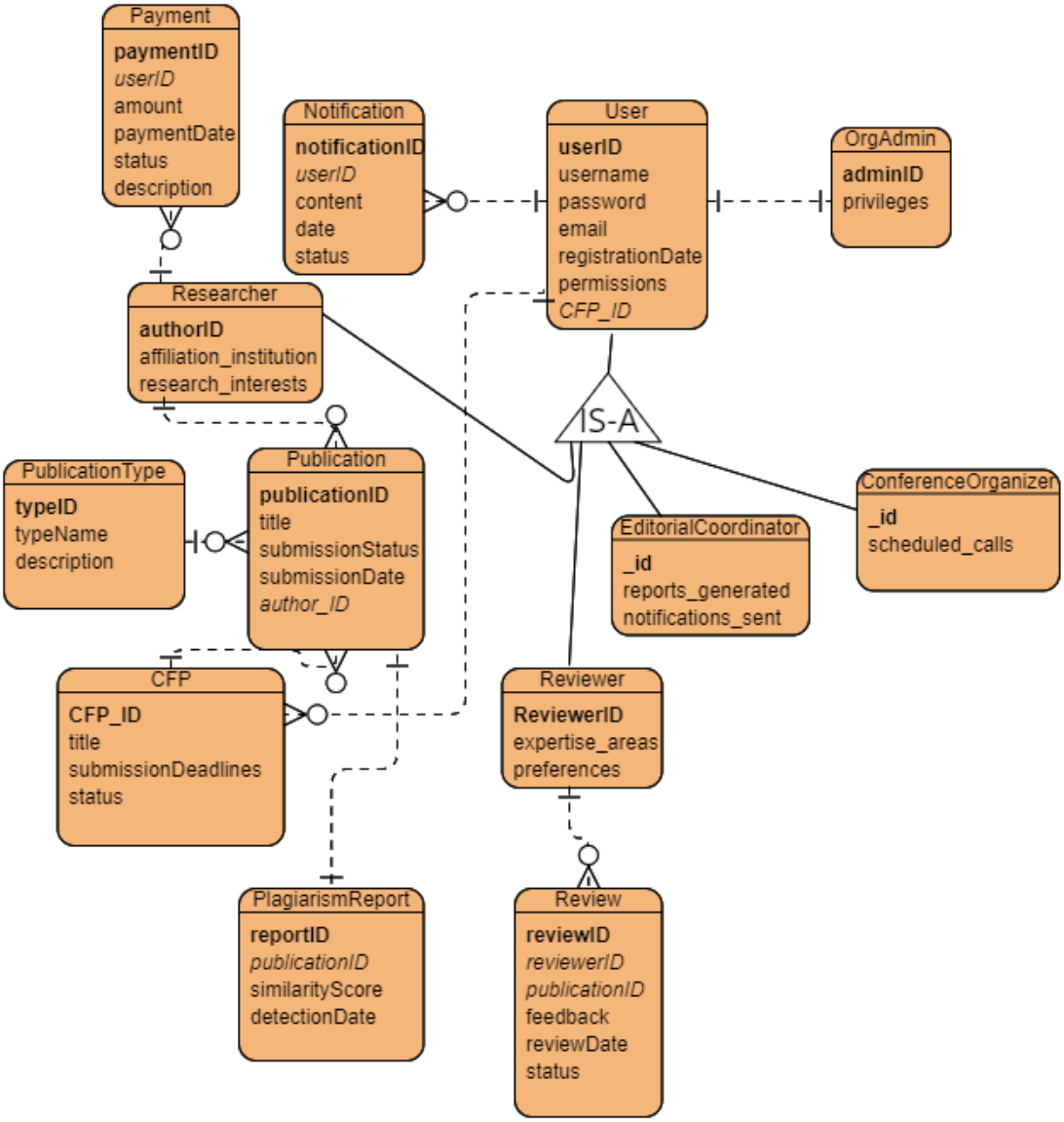

Figure 14: Entity Relationship Diagram

The Data Model is defined as an abstract model that organizes data description, data semantics, and consistency constraints of data. Present appropriate conceptual, logical and physical data models using ER diagrams or UML class diagram.

4.7 User Interface Design

4.7.1 Landing Page

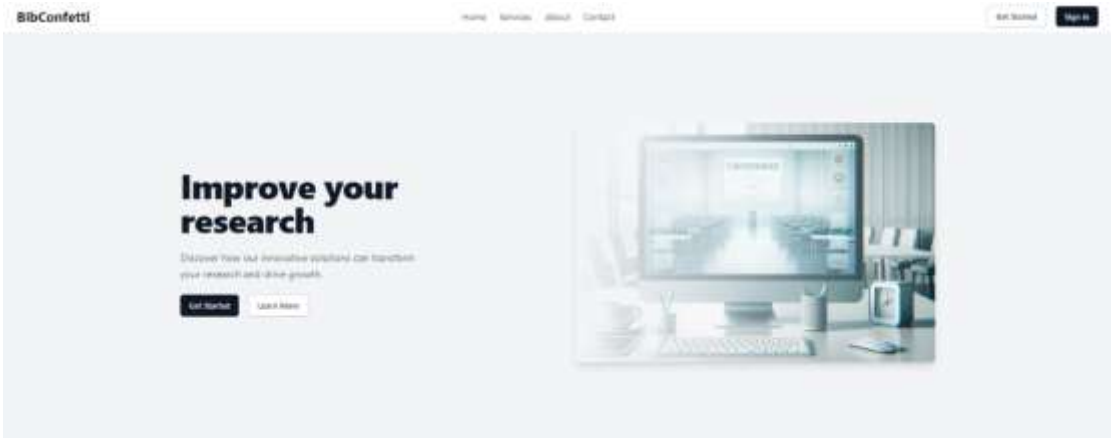

Figure 15: Landing Page

4.7.2 Login

Figure 16: login

4.7.3 Signup

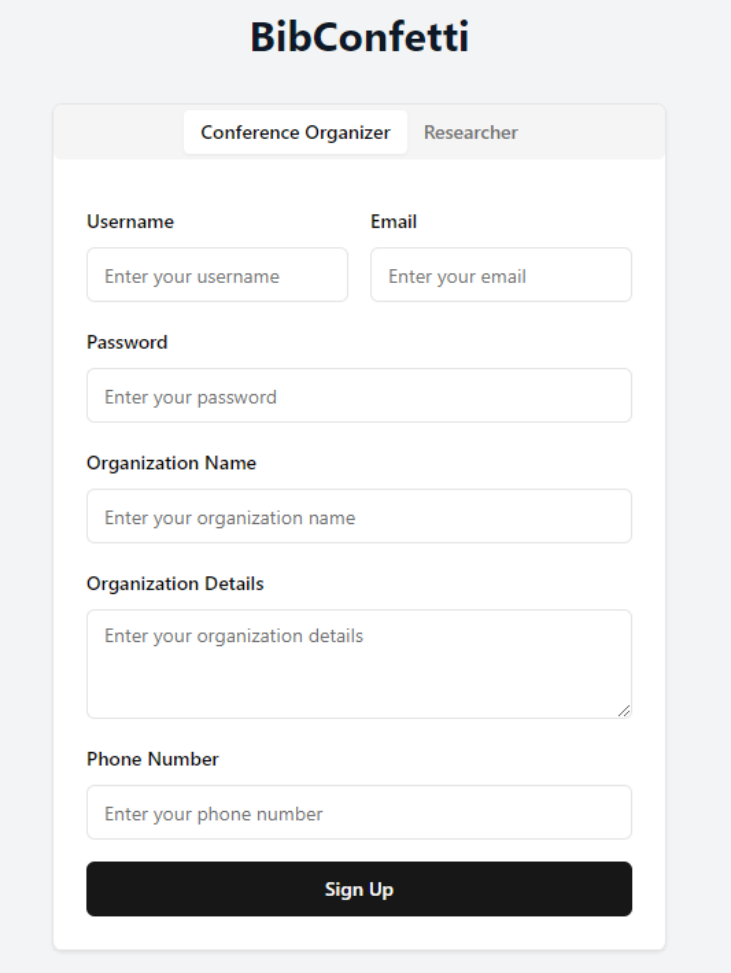

Figure 17: SignUp

4.7.4 Organizer Dashboard

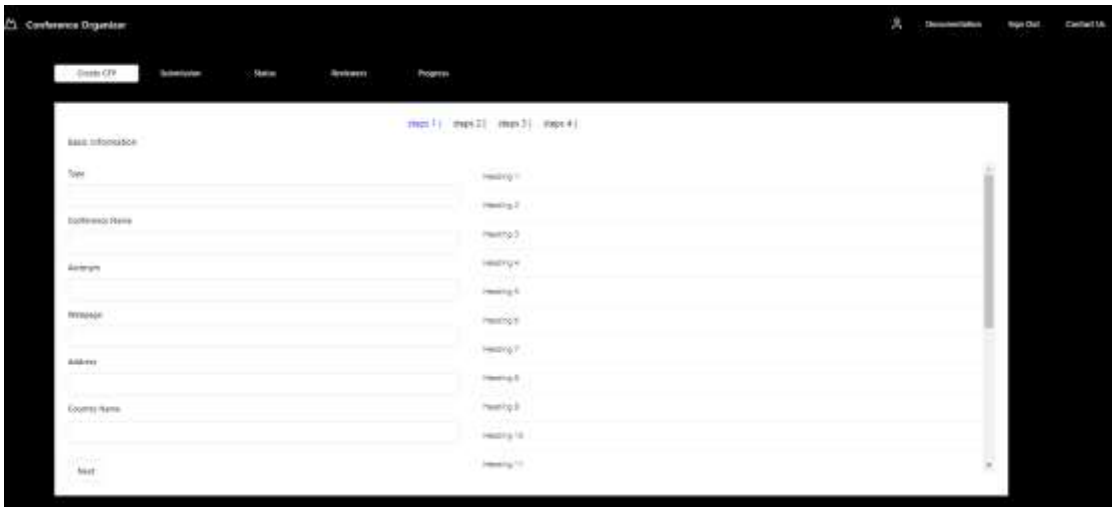

Figure 18: Dashboard

4.7.5 Abstract:

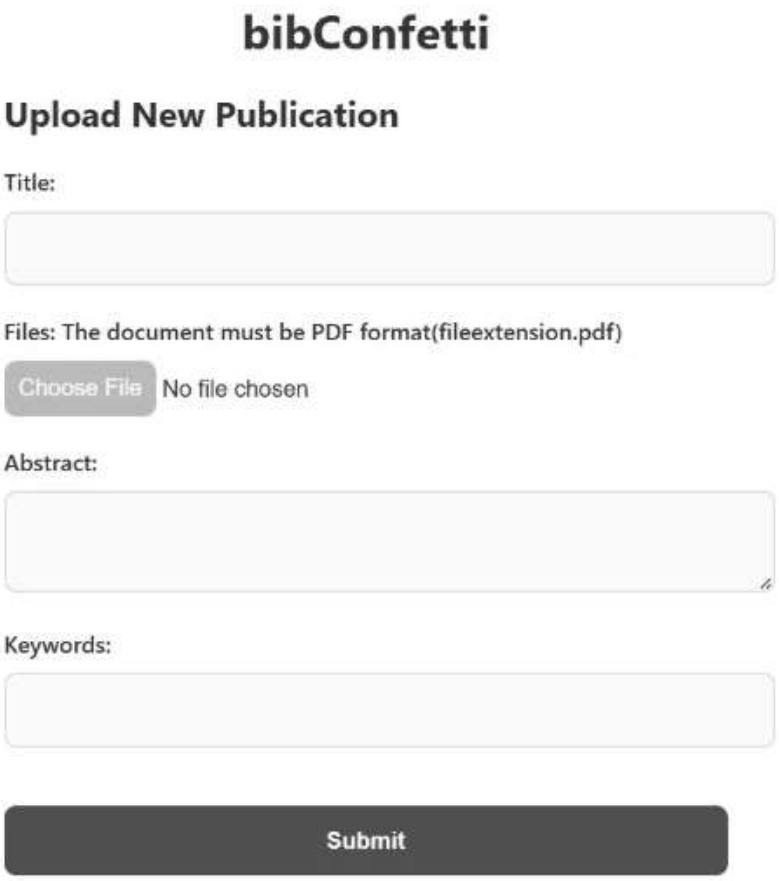

Figure 19: Abstract

(low-fidelity)

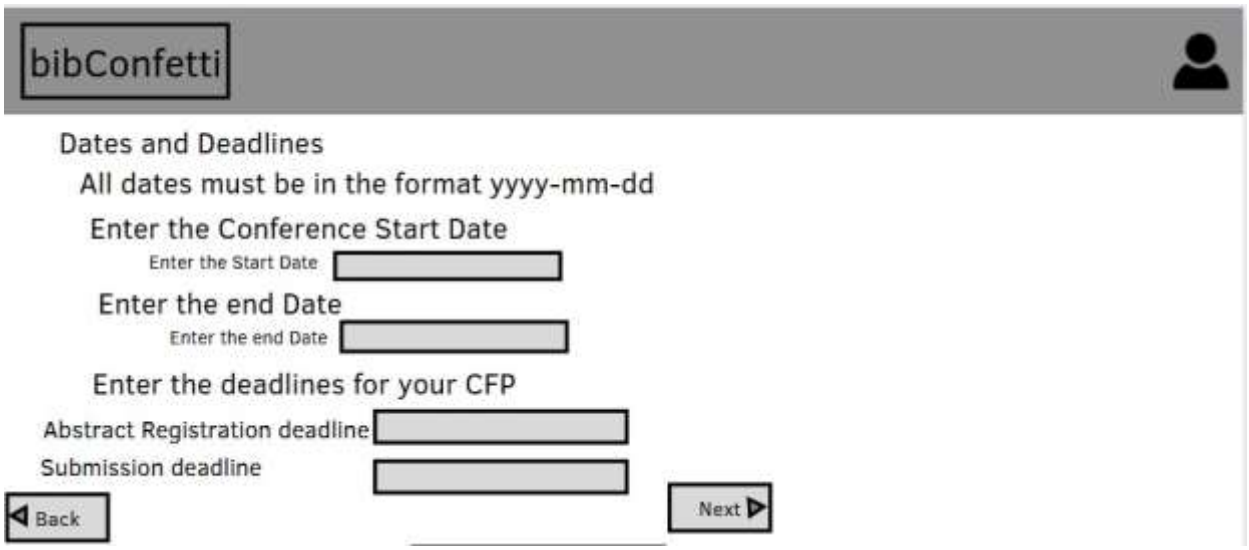

Figure 20: Deadlines

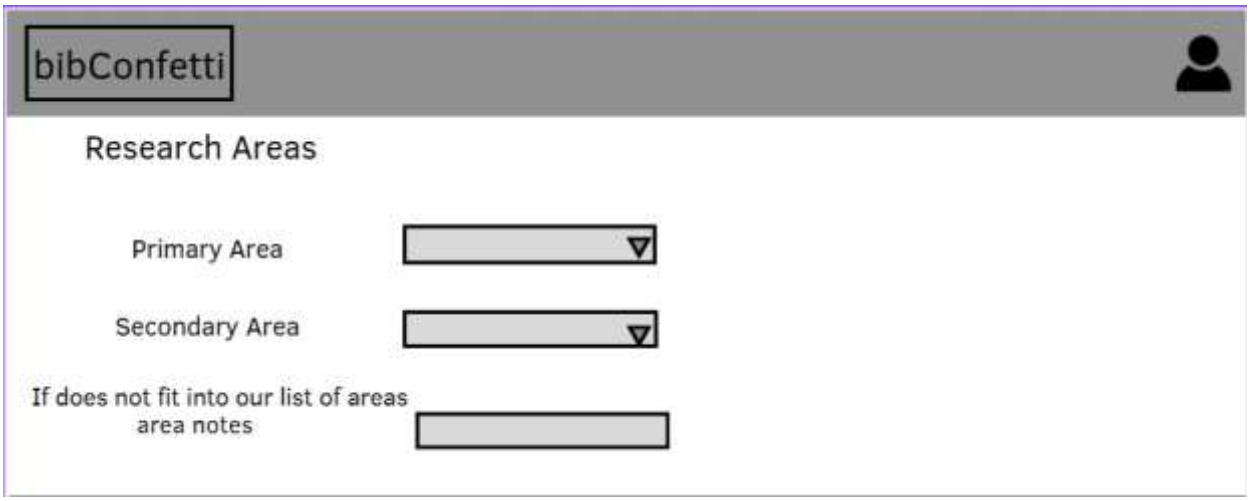

Figure 21: Research Areas

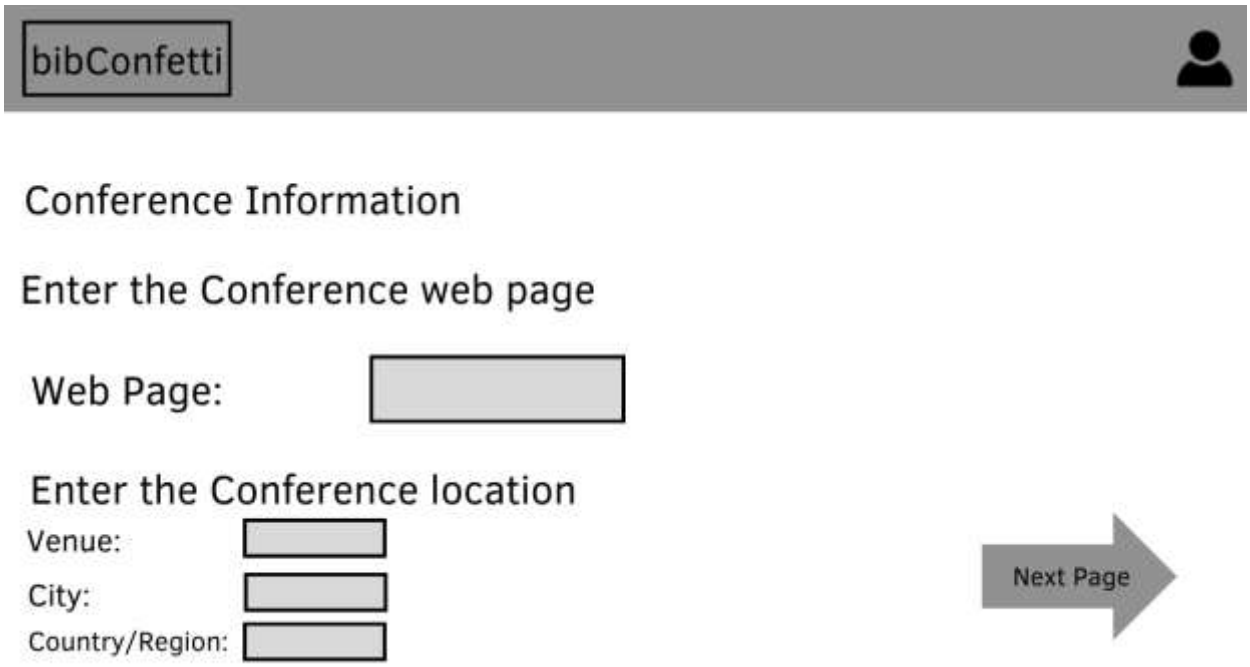

Figure 22: Conference information

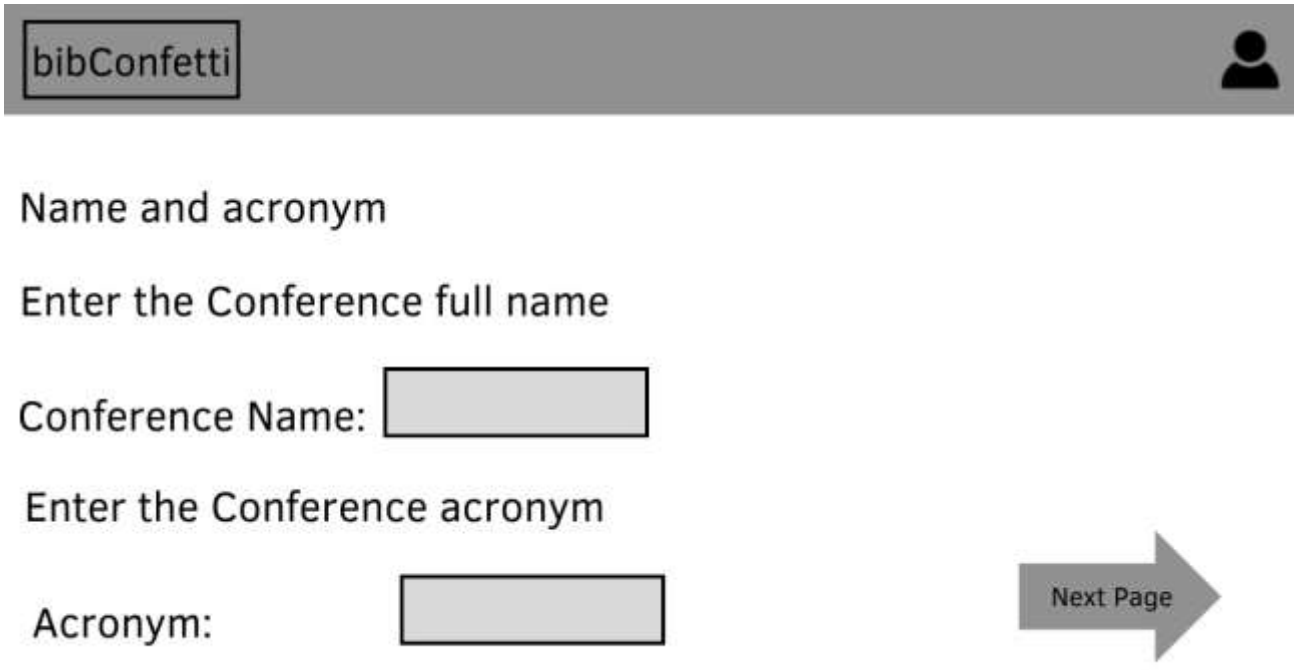

Figure 23: Conference

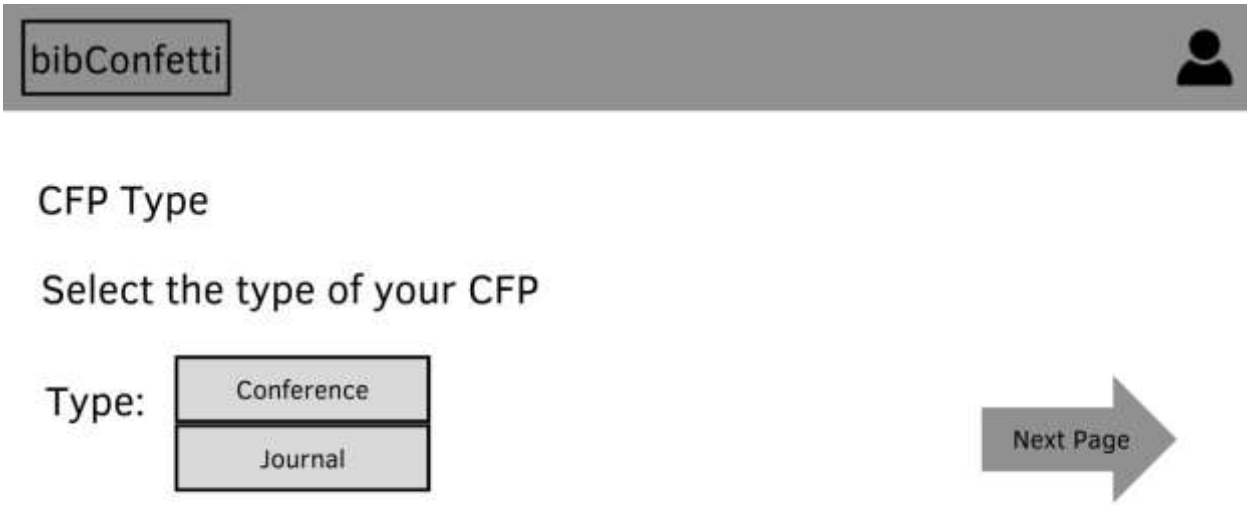

Figure 24: Type

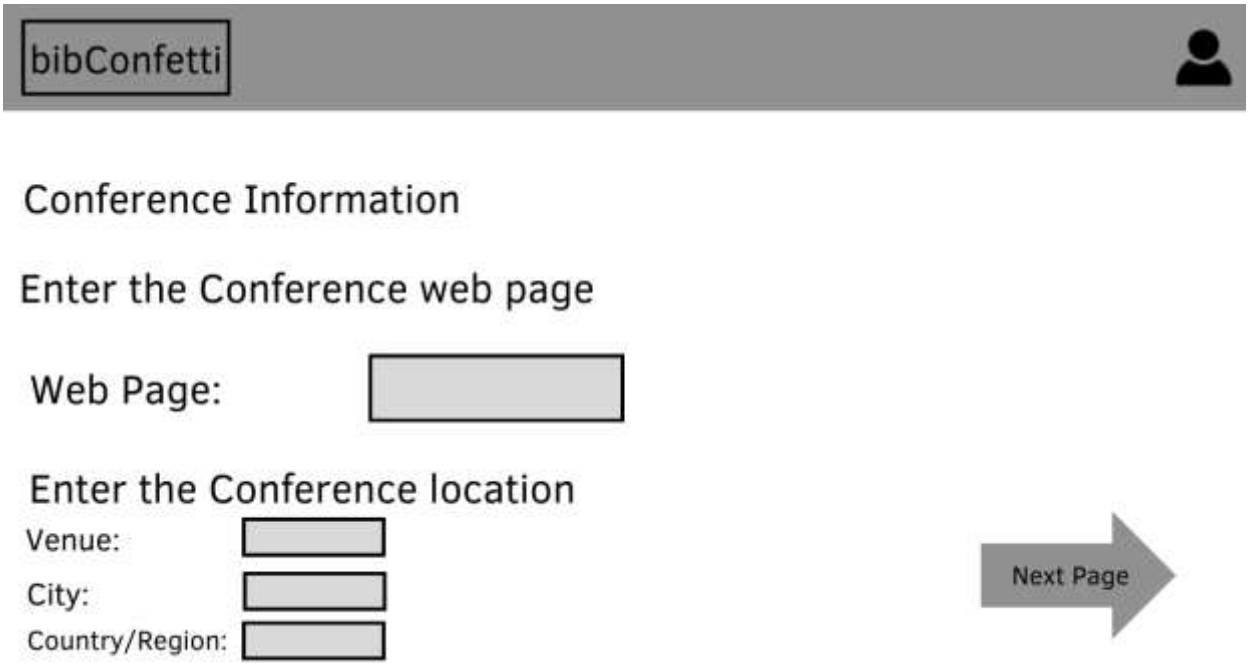

Figure 25: Conference detail

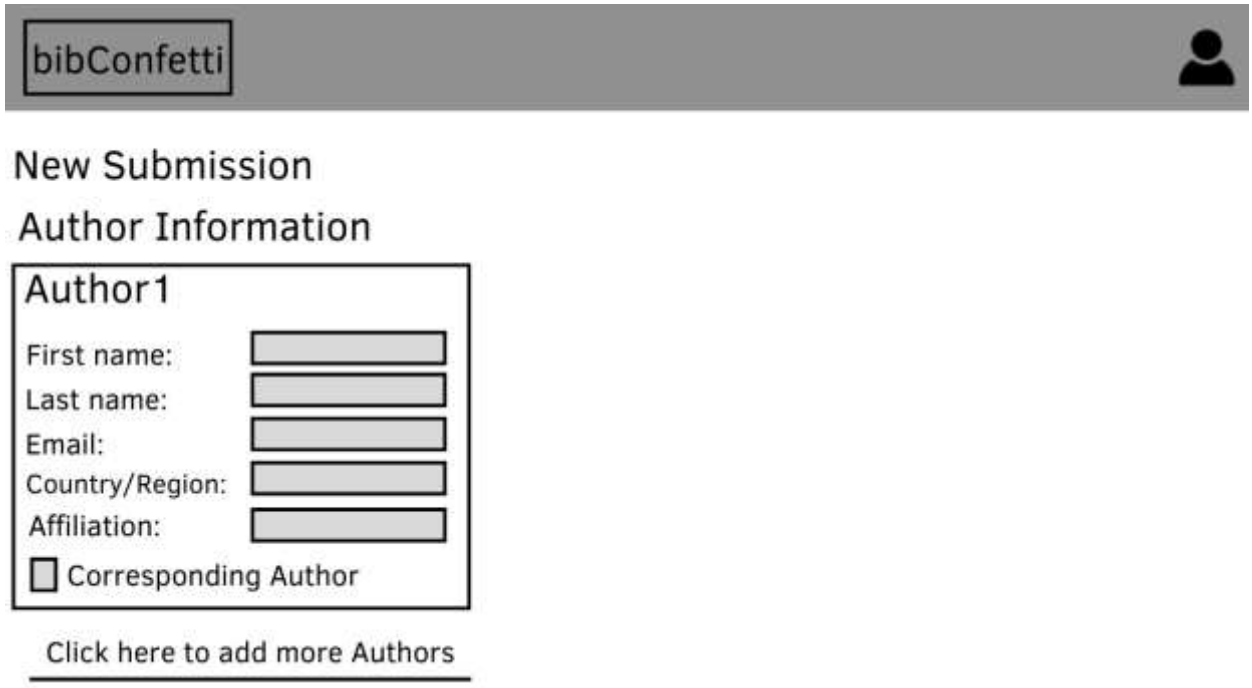

Figure 26: Author

Chapter 5
System Implementation

This chapter describes the implementation phase of our web Application. In this section, Tools and techniques used in the development and completion of our project are explained.

5.1 Tools and Technologies:

In this section, Tools and techniques used in the development and completion of our project are:

1) Tools

Documentation Microsoft Word, Microsoft PowerPoint, .Pdf files.

VS Code, used for coding frontend and backend also.

Visual Paradigm UML Diagrams

GitHub Development vision collaboration platform

2) Technology

React.js/Next.js for developing frontend.

Node.js for developing Backend.

MongoDB for database

AI-NLP Large Language models for AI features Implementation.

User Interfaces

The user interfaces for the Academic Conference & Research Article Management System will be designed to provide an intuitive and efficient user experience for various user classes. Sample screen images and layouts will be presented in the user interface specification document, ensuring consistency and adherence to GUI standards. Standard buttons such as "Submit," "Update," and "Withdraw" will be uniformly implemented across interfaces. Keyboard shortcuts will be incorporated where applicable. Error messages will follow a standardized format for clarity, and a comprehensive help function will be available.

Hardware Interfaces

The software is designed to operate on standard computing hardware, including desktops, laptops, and servers. It does not impose specific hardware requirements beyond those expected for modern web applications. The system will interact with hardware components for data storage and retrieval. Hardware interfaces are standard and do not necessitate specialized protocols.

Software Interfaces

The system interacts with a NOSQL database of MongoDB for data storage and retrieval. The user interfaces are accessed through standard web browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari. Integration with a third-party plagiarism checking service for validating the authenticity of submitted content.

It also includes interaction with a payment processing service for handling financial transactions. The web application is developed using specific server-side technologies of Node.js.

Data sharing mechanisms are standard, utilizing the database for persistent storage and retrieval and data items shared across software components include user credentials, conference details, and publication information.

Communications Interfaces

The communication interfaces in this system are designed around the MERN (MongoDB, Express.js, React, Node.js) stack and incorporate Python for enhanced functionality. The user interface, developed using React, provides a dynamic and responsive front-end experience. System-to-system communication relies on RESTful APIs implemented with Express.js on the Node.js runtime, facilitating efficient data exchange between the React-based front-end and the Node.js-based back-end. MongoDB, accessed through the Mongoose library, serves as the primary database. Python is seamlessly integrated into the system, augmenting specific functionalities. It communicate with the main system components, providing additional capabilities such as AI-driven interactions or specialized data processing. External services, including payment gateways, are integrated through specific APIs. The communication protocols prioritize security, with HTTPS employed for encrypted data transmission. JSON remains the primary data format for structured interchange.

Conclusion

In this chapter tools and technologies which are used to develop our project are mentioned. Basically, we use MERN stack for development of our project which consists of React.js for developing frontend, Node.js for developing Backend, MongoDB for database which helped us a lot to get our required application.

Chapter 6
System Testing & Evaluation

In this chapter we will explain and check whether our system is working properly or not. We will test each scenario against its expected outcome and actual output. Basic Aim of this phase is to check whether our system is error free, and quality of system is maintained or not and to fix such problems before handing over to client.

6.1 Test Strategy

Testing is one of the basic steps in development of a software in this phase we check weather our system is working properly or not. We use different testing techniques to ensure that our system is fulfilling the requirements, and we are getting our required output or not and check some other qualities like speed, response performance etc to check how our system is working and to check whether our system is error free, and quality of system is maintained or not and to fix such problems. Firstly, we code a module and perform a test on it to ensure that the module is working properly or not and then we implement this module in the overall system and perform more tests so that its functionality is assured when used with the overall system. Following are some different techniques that we use in testing phase:

6.2 Component Testing

In this type of testing, we thoroughly check our modules one by one, and we make sure that every module is working properly and giving its outcome properly or not. Component testing precedes integration testing, ensuring that each module functions independently as expected and meets the requirements. We check correctness and verify functionalities that each module is performing before doing their integration.

6.3 Unit Testing

In this testing we check smaller units by using use case testing so that we check working of these units relevant to the required requirements and check weather these units or smaller components of our module are working properly or not. We can also choose manual or automated methods for testing of these units.

6.4 Integrated Testing

In this type of testing, we check the correctness of the module's functionality of the system is tested by integrating different modules of the system. As different modules are interconnected with each other, so these are tested with different inputs and their actual results are compared with the expected output to verify correctness. After the completion of integration testing, we move to next step where the system testing is performed.

6.5 System Testing

System testing include testing whole system and check whether all requirements are fulfilled or not. We use black box testing technique so we get accurate and more precise results, and we can conclude that from this testing all our requirements are fulfilled, and our system is giving the required output.

6.6 Test Cases

Following are some test cases which are showing working of different scenarios:

6.6.1 User Management

6.6.1.1 Test Case No. 1

Table 24: Test case of Registration by Email

Test case	01
Objective	Registration By Email
Pre-Condition	1. Good Internet Connection as it is a Web Application.
	2. The User adds information to register.

Flow	1. User Open Web Application. 2. The User enters their Email and Password. 3. The User checks the Terms and Condition dialog box. 4. The user clicks on the register button. 5. System Verify Entered email and Register the User. 6. System Display Message that "Your Account Has been. Registered Successfully".
Expected Output	Registration Successful
Actual Output	Registration Successful
Status	The Test Was Successfully Performed.

6.6.1.2 Test Case No. 2

Table 25: Test case of Login

Test case	02
Objective	Login Screen
Pre-Condition	The User adds information to the login.

Flow	1. Enter the email and password. 2. Click on the login button. 3. System Display Message That "Your you have login Successfully".
Expected Output	Login Successfully and the home page is displayed.
Actual Output	Login Successfully and the home page is displayed.
Status	The Test Was Successfully Performed.

6.6.1.3 Test Case No. 3

Table 26: Test case of Forgot password.

Test case	03
Objective	Forgot Password
Pre-Condition	The web app is open, and the user is on the login page.

Flow	1. Enters email and incorrect password. 2. Click Login. 3. The system display's invalid username or password error. 4. The user clicks on forgot password button. 5. Forget the password page open in which users must enter their email address and then click on the Reset button. 6. User opens their email account in which user found a mail through which they can reset their password. 7. The user opens that mail and changes the password and clicks on save. 8. System display message that "Password Reset successfully".
Expected Output	Password changed successfully.
Actual Output	Password changed successfully.
Status	The Test Was Successfully Performed.

6.6.1.4 Test Case No. 4

Table 27: Test case of Logout

Test case	04
Objective	Logout
Pre-Condition	1. The user is logged in.

	2. The user no longer wants to be logged in.
Flow	1. The user is done using the web application. 2. The user clicks on the logout button. 3. The system logs the user out.
Expected Output	Logout successfully.
Actual Output	Logout successfully.
Status	The Test Was Successfully Performed.

6.6.1.5 Test Case No. 5

Table 28: Test case of upload publication

Test case	05
Objective	Select Publication
Pre-Condition	The web app is open, CFP exists, abstract approved and the user is at the upload publication page.
Flow	1. User selects the upload publication button. 2. User clicks on the submit button and upload publication.
Expected Output	Publication is uploaded successfully.
Actual Output	Publication is uploaded successfully
Status	The Test Was Successfully Performed.

6.6.1.6 Test Case No. 6

Table 29: Test case of search publication

Test case	06
Objective	Search Publication
Pre-Condition	The web app is open, and the user is at the dashboard.
Flow	1. User clicks on the search publication button. 2. Users enter the publication name and select required publication from list if search matches.
Expected Output	Publication was founded successfully.
Actual Output	Publication was founded successfully.
Status	The Test Was Successfully Performed.

6.6.1.7 Test Case No. 7

Table 30: Test case of feedback

Test case	07
Objective	Feedback to publisher from reviewer
Pre-Condition	Reviewer gives his review for a publication.
Flow	1. Reviewer gives feedback for a publication. 2. The Publisher was notified and able to check and read comments from reviewer's feedback.
Expected Output	Feedback was submitted successfully.
Actual Output	Feedback was submitted successfully.
Status	The Test Was Successfully Performed.

6.6.1.8 Test Case No. 8

Table 31: Test case of Delete from Publications.

Test case	08
Objective	Delete Publication.
Pre-Condition	The web app is open, and the user is at the home page.
Flow	1. User selects the publication and clicks on the delete icon. 2. The selected publication is deleted from the individual submitted section.
Expected Output	Publication deleted from the individual submitted section successfully.
Actual Output	Publication deleted from the individual submitted section successfully.
Status	The Test Was Successfully Performed.

6.6.1.9 Test Case No. 9

Table 32: Test case of View Publication

Test case	09
Objective	View Publication
Pre-Condition	Publisher opens dashboard where list of his publications is show.
Flow	1. User clicks on the publication and able to view the publication.
Expected Output	Publication was Viewed successfully.
Actual Output	Publication was Viewed successfully.
Status	The Test Was Successfully Performed.

6.6.1.10 Test Case No. 10

Table 33: Test case of Chat with publication

Test case	10
Objective	Chat With Publication
Pre-Condition	The web app is open, and the user is at the dashboard.
Flow	1. User First open a publication and click on chat with pdf feature and able to chat with document which was created using AI.
Expected Output	User can Chat with Publication successfully.
Actual Output	User can Chat with Publication successfully.
Status	The Test Was Successfully Performed.

6.6.2 Conference Organizer

6.6.2.1 Test case No. 12

Table 34: Schedule Conference

Test case	12
Objective	create call for publication (CFPs)
Pre-Condition	Conference details (date, time, venue) are not scheduled.
Flow	1.The conference organizer initiates the scheduling process, providing necessary details such as date, time, and venue through the system interface. Upon confirmation, the system records and saves the conference schedule.

Expected Output	Conference was scheduled successfully.
Actual Output	Conference was scheduled successfully.
Status	The Test Was Successfully Performed.

6.6.2.2 Test case No. 13

Table 35: Test case of set submission deadline

Test case	13
Objective	Extend the deadline of CFP
Pre-Condition	Conference details are available.
Flow	1. Accessing the conference management system, navigating to the submission deadline settings, and defining the deadlines for authors to submit their publications.
Expected Output	Deadline was set successfully.
Actual Output	Deadline was set successfully.
Status	The Test Was Successfully Performed.

6.6.2.3 Test case No. 14

Table 36: Test case Publish accepted publications.

Test case	14
Objective	Publish accepted publications.
Pre-Condition	Accepted submissions have been finalized and payments have been done.

Flow	1.Conference Organizer selects the accepted submissions for publication. System processes the selected submissions for publishing.
Expected Output	Publications uploaded successfully.
Actual Output	Publications uploaded successfully.
Status	The Test Was Successfully Performed.

6.6.3 Reviewing Committee

6.6.3.1 Test case No. 15

Table 37: Test case of Select a publication for review.

Test case	15
Objective	Select publication for review.
Pre-Condition	The system has assigned publications for review using AI-NLP.
Flow	1. The Reviewing Committee, after logging in, selects a publication from the list of assigned publications. The chosen publication is marked for review.
Expected Output	Publication was selected successfully.
Actual Output	Publication was selected successfully.
Status	The Test Was Successfully Performed.

6.6.3.2 Test Case No. 16

Table 38: Test case of feedback

Test case	16
Objective	Feedback to publisher from reviewer
Pre-Condition	Reviewer gives his review for a publication.
Flow	1. Reviewer gives feedback for a publication. 2. The Publisher was notified and able to check and read comments from reviewer's feedback.
Expected Output	Feedback was submitted successfully.
Actual Output	Feedback was submitted successfully.
Status	The Test Was Successfully Performed.

6.6.4 Organizational admin

6.6.4.1 Test case No. 17

Table 39: Test case of assigning roles.

Test case	17
Objective	Manage User Accounts (User Roles and Permissions)
Pre-Condition	The admin adds information to the login.
Flow	1.Organizational Admin logs in and accesses the user management section. 2.Organizational Admin manages user accounts by assigning roles and permissions, creating new accounts, or modifying existing ones by communicating through emails by providing their relevant account's information.

Expected Output	Login Credentials were sent to users.
Actual Output	Login Credentials were sent to users.
Status	The Test Was Successfully Performed.

6.6.4.3 Test case No. 18

Table 40: Test case of Handling Payment

Test case	18
Objective	Handling Payment
Pre-Condition	Organizational Admin is authenticated, payment processing module is operational
Flow	1. Organizational Admin logs in and accesses the payment management section. 2.Organizational Admin reviews and processes payments, ensuring all financial transactions are accurate and complete.
Expected Output	Password changed successfully.
Actual Output	Password changed successfully.
Status	The Test Was Successfully Performed.

6.6.4.4 Test case No. 18

Table 41: Test case of Logout

Test case	18
Objective	Logout
Pre-Condition	1. The user is logged in. 2. The user no longer wants to be logged in.
Flow	1. The user is done using the mobile application. 2. The user clicks on the logout button. 3. The system logs the user out.
Expected Output	Logout successfully.
Actual Output	Logout successfully.
Status	The Test Was Successfully Performed.

Results & Evaluation

The results are: Development of the system: We were able to successively develop BibConfetti, it has such features as user registration ability, login process, submission of both papers, journals, and articles, automatic reviewer allocation via AI among others like conversations with AI chatbots through it, AI analyses extraction of key insights, interfacing to a plagiarism prevention program, processing of paper submission charges using money transfer services, searching facilities and even deadline notifications and status updates as well as for browsing purposes.".

Conclusion

In this chapter we performed different types of testing to check correctness of our project according to given requirements we also make test cases and check possible outcomes on these units and check their outcomes.

Chapter 7
Conclusion

The BibConfetti represents a new, self-contained product designed to improve the management of academic conferences and research article submissions. It is not a replacement for existing systems but rather a standalone solution catering to the specific needs of academic institutions and researchers. The system operates independently but may interface with external services for plagiarism checking and payment processing. While it functions autonomously, its seamless integration with external systems ensures a holistic approach to conference management and publication workflows.

7.1 Contributions

The scope of our contribution is fully within the framework of the research objectives we pursue. Initially, we are going to put in place BibConfetti, a system for managing conferences that will be AI driven and directly deal with the challenge of simplifying review processes by automating the tasks such as reviewer assignment and plagiarism detection. The second point to be made here is the attention we pay to the intuitive interface design and interactive features. Through that, we improve the user experience and engagement and, therefore, attain the goal of upskilling usability. Finally, our contributions do not only provide specific responses to the research questions within the topic of academic conference management, but they also carry out advancements of the knowledge and resolution of practical problems in this domain of activity.

7.2 Reflections

7.2.1 Advantages

1. A positive and clear UI that people adore. At the same time, it must be useful and understandable.

2. At the application's core, automation is a key ingredient, which reduces the majority of tasks involving clicks just to be one or two.

3. The service is, however, highly simplistic to use; hence no education whatsoever is required.

7.2.2. Drawbacks: In parallel, it is possible to purchase and experience these art pieces through online and in some cases offline platforms at this time.

7.2.3 Impact on Society/Knowledge

Useful Applications: Researchers, meetings organizers as well as the academic community will benefit because BibConfetti will serve them in terms of efficiency and accessibility besides increasing transparency of academic conference in general.

Adding to the Body of Knowledge: BibConfetti development and its evaluation at the same time teach us more about human-oriented design, academic artificial intelligence initiatives, and engagement opportunities for conference managers.

Overcoming Obstacles: Through applying the solutions against the major aspects of conference management such as cumbersome manual operations.

7.3 Future work

7.3.1 Other platforms

BibConfetti web app has been developed for web platform. In future it can be designed on android and iOS-based mobile devices too.

REFERENCES

[1]

Gupta, Bharat & Gupta, O P & Sawhney, BK. (2013). A Framework for Conference Management System. International Journal of Management & Information Technology. 5. 483-487. 10.24297/ijmit.v5i2.764.

[2]

Tewari, Tribhuwan. (2010). Survey of Conference Management Systems. International Journal of Computer Applications. 2. 14-20. 10.5120/632-875.

[3]

Kou, Ngai Meng & Uu, Ryan Leong Hou & Mamoulis, Nikos & li, Yuhong & Li, Ye & Gong, Zhiguo. (2015). A topic-based reviewer assignment system. Proceedings of the VLDB Endowment. 8. 1852-1855. 10.14778/2824032.2824084.

[4]

Kalmukov, Yordan. (2011). Architecture of a Conference Management System Providing Advanced PaperAssignment Features. International Journal of Computer Applications. 34. 51-59. 10.5120/4083-5888.

[5]

Wan Ishak, Wan Hussain & Mat Yamin, Fadhilah & Mohsin, Mohamad & Mansor, Mohd Fitri. (2023). A COMPARATIVE REVIEW OF CONFERENCE MANAGEMENT SYSTEM. Journal of Technology and Operations Management. 18. 87-93. 10.32890/jtom2023.18.2.7.

[6]

Deutschland, Aboli & pune, Parikshit & MCERC, Varsha & MCERC, Swati. (2022). An Efficient and Intelligent Conference Management System- Desired Services and Features. 10.21203/rs.3.rs-1900925/v1.

[7]

Lubis, Muharman & Zunaedi, Iqbal & Musnansyah, Ahmad & Fauzi, Rahmat. (2022). Design Approach in Conference Management System with EZDESK Dashboard for Digital Ecosystem. 1-7. 10.1109/ICOSTECH54296.2022.9829166.

[8]

Ferilli, Stefano & Di Mauro, Nicola & Basile, Teresa & Esposito, Floriana & Biba, Marenglen. (2006). Automatic Topics Identification for Reviewer Assignment. 4031. 721-730. 10.1007/11779568_78.

[9]

Huang, Min & Feng, Yuwei & Desai, Bipin. (2009). ConfSys2: an improved web-based multi-conference management system. 155-159. 10.1145/1557626.1557651.

[10]

Stadt, Richard. (2012). CyberChair: A Web-Based Groupware Application to Facilitate the Paper Reviewing Process.