

**Bilkent University**  
**Department of Computer Engineering**



**Senior Design Project**

**T2409**

Conferencer

**Final Report**

**02.05.2025**

**22003569, Abdurrahman Bilal Kar, bilal.kar@ug.bilkent.edu.tr**

**22003153, Ahmet Memduh Tutuş, memduh.tutus@ug.bilkent.edu.tr**

**22103189, Atilla Emre Söylemez, emre.soylemez@ug.bilkent.edu.tr**

**22003111, Berkay Ayçiçek, berkay.aycicek@ug.bilkent.edu.tr**

**Supervisor: Can Alkan**

**Instructors: Mert Bıçakçı, Atakan Erdem**

**Innovation Expert: Muhammed Naci Dalkıran**

1. Introduction	4
2. Requirements Details	5
2.1 Functional Requirements	5
2.1.1 User Authentication and External Integrations	5
2.1.2 Program Committee Management	6
2.1.3 Conference Creation and Call for Papers	7
2.1.4 Paper Submission & Tracking	8
2.1.5 Multi-track Handling	9
2.1.6 Assigning Reviewer System	10
2.1.7 Reviewing Assigned Papers	11
2.1.8 User Profile Page	12
2.1.9 LLM-Generated Review Detection (Future Plan)	13
2.2 Non-functional Requirements	14
2.3 Pseudo Requirements	15
3. Final Architecture and Design Details	16
3.1 Software Architecture	16
3.1.1 Overview	16
3.1.2 Subsystem Decomposition	16
3.1.3 Hardware/software mapping	17
3.1.4 Persistent data management	18
3.1.5 Access control and security	18
3.2 Subsystem Services	18
3.2.1 Authentication and Access Control Service	18
3.2.2 Conference Management Service	19
3.2.3 Submission Management Service	19
3.2.4 Review Assignment and Management Service	19
3.2.5 Decision-Making Service	20
3.2.6 User Profile and Analytics Service	20
3.2.7 Notification and Communication Service	20
4. Development/Implementation Details	21
4.1 Technology Stack	21
4.2 Frontend Implementation	22
4.3 Backend Implementation	23
4.4 Database Design	23
4.5 Access Control and Security	24
4.6 CI/CD and Deployment	24
5. Testing Detail	25
<b>6. Maintenance Plan and Details</b>	<b>56</b>
6.1 System-Level Maintenance	57
6.2 Application Stack Management (React, Flask, Nginx, MongoDB)	57
6.3 Database Maintenance (MongoDB)	58
6.4 Monitoring, Logging, and Alerting	58
6.5 Security	59
6.6 Version Control and Deployment	59

7. Other Project Elements	60
7.1 Consideration of Various Factors in Engineering Design	60
7.1.1 Constraints	62
7.1.2 Standards	64
7.2 Ethics and Professional Responsibilities	65
7.2.1 Ethical Requirements	65
7.2.2 Professional Requirements	65
7.3 Teamwork Details	65
7.3.1 Contributing and functioning effectively on the team	65
7.3.2 Helping creating a collaborative and inclusive environment	66
7.3.3 Taking lead role and sharing leadership on the team	67
7.3.4 Meeting objectives	68
7.4 New Knowledge Acquired and Applied	70
8. Conclusion and Future Work	73
9. Glossary	74
10. References	74

# 1. Introduction

In the rapidly evolving landscape of academic conferences, the process of managing submissions, reviews, and program committees has become increasingly complex and time-consuming. Organizers and reviewers are often tasked with managing large volumes of data, ensuring fair and efficient review processes, and maintaining a high standard of academic integrity. Existing systems such as EasyChair, HotCRP, and OpenConf, while useful, fail to address the nuanced needs of modern conferences, particularly in terms of usability, flexibility, and integration with other academic platforms. Our purpose is to fulfill the lack of a comprehensive, user-friendly, and flexible conference management system.

Our proposed system, Conferencer, is designed to regularize and enhance the conference management process for academic events. It offers a centralized platform where organizers, reviewers, and authors can interact, collaborate, and efficiently manage the submission, and review processes. By focusing on user experience and integration with external platforms like ORCID, Web of Science, and Google accounts, the system aims to provide a flawless, intuitive, and secure environment for all stakeholders involved in the conferences.

The system is a set of carefully crafted functional requirements that address the various stages of the conference process, from initial conference setup to paper submission and reviewer management. The platform is designed to be flexible, allowing organizers to configure their conference events, define tracks, and manage paper submissions with ease. For authors, it provides a useful interface for submitting papers, tracking their status, and receiving feedback. Reviewers can have the benefit of a recommendation engine that matches them with papers based on their expertise, past performance, and conflict-of-interest declarations.

Role-based access control within the conference system ensures that users, whether they are authors, reviewers, or admins, have access to only the features relevant to their roles. Organizers can manage active sessions, set up auto-logout for inactive users, and ensure that the entire conference management process is secure and well-organized.

The system also focuses on performance and usability for program committee (PC) management. Organizers can invite potential PC members, and track the status of invitations. The system's intelligent search and filtering features help identify optimal reviewers for papers, and its historical performance data assists in making informed decisions about reviewer assignments. For multi-track conferences, the system provides dedicated tools for managing each track independently, allowing for customized workflows while maintaining confidentiality across tracks.

A key focus of the system is its reviewer management and performance evaluation tools. Reviewers can customize their profiles, declare conflicts of interest, and set preferences for the types of papers they wish to review. Conflict-of-interest detection system automatically flags potential issues by cross-referencing reviewer profiles with affiliations and co-authorships. Track and conference chairs can rate reviews based on quality, and reviewers can also rate their peers to ensure the integrity and fairness of the review process.

Looking to the future, Conferencer plans to incorporate advanced technologies like Natural Language Processing (NLP) to detect reviews generated by large language models (LLMs). This will help to maintain the authenticity of the review process and prevent the use of automated tools to generate reviews that lack the depth and engagement expected of academic peer reviews. By integrating such tools, the system will ensure that the credibility and trust of the review process are maintained, while also enhancing the user experience for both authors and reviewers.

## **2. Requirements Details**

### **2.1 Functional Requirements**

#### **2.1.1 User Authentication and External Integrations**

- **2.1.1.1 User Authentication**

- Support login via native accounts (email and password).
- Enable authentication through third-party services like:
  - Google accounts.
  - ORCID accounts.

- **2.1.1.2 Profile Management**

- Allow users to create and manage their profiles securely.
- Provide functionality for updating personal details such as name, email, affiliation, and contact information.

- **2.1.1.3 Reviewer Integration with ORCID**

- Allow reviewers to link their profiles with their ORCID accounts.
- Fetch and display verified reviewer information from ORCID (e.g., published works, affiliations).

- **2.1.1.4 Web of Science Integration**

- Integrate with the Web of Science platform for:
  - Automatically recognizing peer review contributions.
  - Providing external validation for reviews to enhance credibility.
- Allowing users to export their review history for professional or academic purposes.

- **2.1.1.5 Session Management**

- Allow users to view and manage active sessions for their accounts.
- Implement auto-logout for inactive sessions to enhance security.

- **2.1.1.6 User Access Control**

- Define role-based access for different user types:
  - Authors: Manage submissions and profile settings.
  - Reviewers: View assigned papers, submit reviews, and manage their profiles.
  - Admins: Configure settings, manage users, and moderate content.

## **2.1.2 Program Committee Management**

- **2.1.2.1 Program Committee Invitations and Importing**

- Allow organizers to send personalized invitations to potential PC members.
- Enable bulk invitations via email or integrated platforms like ORCID or institutional directories.
- Track the status of invitations (e.g., sent, accepted, declined).
- Provide functionality to import PC members from:
  - Previous years' committees.
  - External files (e.g., CSV, Excel).
- Allow bulk editing or updating of imported member details.

- **2.1.2.2 PC Member Search, Filtering, and Assignment**

- Enable search functionality for PC members using:
  - Keywords (e.g., expertise, affiliation, research area).
  - Past review scores or feedback.
  - Availability status or number of assigned papers.
- Suggest optimal reviewers for papers based on research interests, past performance, or conflicts of interest.

- **2.1.2.3 Conflict of Interest and Role Management**

- Allow PC members to declare conflicts of interest, with automatic detection based on affiliations or co-authorship.
- Support role-based permissions (e.g., senior reviewer, chair) for efficient committee management.
- Enable organizers to define different roles for PC members (e.g., senior reviewer, junior reviewer, chair).
- Assign specific permissions based on roles (e.g., paper reassignment for chairs).

- **2.1.2.4 Reviewer Performance and Historical Data**

- Track PC member performance metrics such as timeliness, review quality, and engagement.
- Maintain historical data on PC member contributions for future reference and analytics.

- **2.1.2.5 Communication and Notifications**

- Enable internal messaging for announcements, reminders, and direct communication between organizers and PC members.
- Provide email notifications for assignments, deadlines, and updates.

## **2.1.3 Conference Creation and Call for Papers**

- **2.1.3.1 Conference Setup**

- Organizers can create new conferences by specifying key details such as:
  - Conference series (e.g., annual event, specific academic field).
  - Year of the conference.
  - Track structure, allowing multiple tracks for different research areas or themes.

- **2.1.3.2 Track Configuration**

- Organizers can define individual tracks within the conference, including:
  - Track names (e.g., "Artificial Intelligence," "Data Science").
  - Description of the focus or theme for each track.
  - Specific submission criteria and deadlines for each track.

- Ability to modify track details as needed during the setup process.
- **2.1.3.3 Generation of Call for Papers (CFP)**
  - Organizers can easily generate a Call for Papers (CFP) for the conference, including:
    - Overview of the conference, including key dates, themes, and submission guidelines.
    - Track-specific submission details, ensuring potential contributors understand the focus of each track.
    - Submission deadlines, abstract submission windows, and final paper submission requirements.
    - Formatting guidelines, word count limits, and other submission criteria for papers.
- **2.1.3.4 CFP Distribution**
  - Enable organizers to distribute the CFP to potential participants through:
    - Email notifications.
    - Integration with academic mailing lists or online academic networks.

## **2.1.4 Paper Submission & Tracking**

- **2.1.4.1 Intuitive Paper Submission Interface**
  - Authors can easily upload their manuscripts and accompanying materials (e.g., supplementary files, and abstracts) through a user-friendly interface.
  - The interface supports multiple file formats (e.g., PDF, Word, LaTeX) for easy submission.
- **2.1.4.2 Submission Tracking**
  - Authors can track the status of their papers throughout the entire submission process, including:
    - Initial submission.
    - Review stage.
    - Final decision (acceptance, rejection, revisions).
  - The system tracks and manages all critical submission and review deadlines.



- **2.1.4.3 Automated Compliance Checks**

- The system performs automated checks for:
  - Formatting compliance (e.g., font size, margins, title page).
  - File requirements (e.g., maximum file size, acceptable file types).

- **2.1.4.4 Review Assignment and Monitoring**

- Organizers and reviewers can easily assign papers for review based on expertise, availability, or conflicts of interest.
- Reviewers are notified when they have new assigned papers, along with key deadlines.
- Organizers can monitor submission progress and track review completion to ensure timely processing of papers.

- **2.1.4.5 Submission Status Dashboard**

- Provide a dashboard for both authors and organizers to view the overall submission and review status, including:
  - Number of submissions received.
  - Papers under review, awaiting revisions, or completed.
  - Reviewer assignments and feedback.

## **2.1.5 Multi-track Handling**

- **2.1.5.1 Track Support and Management**

- The system accommodates both single-track and multi-track conferences, providing flexibility for different organizational structures.
- Each track in multi-track conferences can be managed independently by its respective track chair, allowing:
  - Separate configurations for submission criteria, deadlines, and review processes.
  - Customization of workflows specific to the needs of each track.

- **2.1.5.2 Shared PC Across Tracks**

- For conferences like the Intelligent Systems for Molecular Biology (ISMB), a shared PC can be utilized across tracks.
- PC members can contribute to multiple tracks while ensuring fair workload

distribution.

- **2.1.5.3 Reviewer Handling**

- PC members can specify the maximum number of papers they are willing to review across all tracks.
- The system enforces these preferences to prevent reviewer overload and ensure equitable distribution of assignments.
- Track chairs are responsible for managing reviewer assignments within their track, ensuring:
  - Confidentiality between tracks, as chairs will not have direct access to assignments in other tracks.
  - Fair allocation of papers to reviewers based on expertise and workload preferences.
- Track chairs can monitor overall reviewer quotas to ensure balanced assignment across tracks.
- Real-time updates help chairs manage resources effectively and avoid delays in the review process.

- **2.1.5.4 Cross-Track Confidentiality**

- Reviewer assignments are kept confidential across tracks, maintaining the integrity and independence of the review process.
- Track chairs can only view aggregate metrics, such as total papers reviewed or pending, across the shared PC.

## **2.1.6 Assigning Reviewer System**

- **2.1.6.1 Reviewer Recommendation Engine**

- The system includes a built-in recommendation engine to streamline the reviewer assignment process.
- Recommendations are based on automated text analysis of submissions, ensuring accurate reviewer-paper matching.

- **2.1.6.2 Similarity Analysis**

- The system extracts relevant keywords from the paper title, abstract, and content.
- Keywords are matched against reviewers' areas of expertise and research interests, ensuring alignment with their knowledge domains.

- The recommendation engine analyzes the similarity between the paper's abstract and reviewers' past publications or expertise.
- This ensures that reviewers are assigned papers closely aligned with their background, improving review quality.
- **2.1.6.3 Reduced Manual Workload for Chairs**
  - The automated recommendations minimize the need for manual reviewer searches, saving time for track chairs and organizers.
  - Chairs can approve, modify, or override recommendations as needed for specific cases.
- **2.1.6.4 Transparent Assignment Process**
  - Track chairs can view a detailed breakdown of why a reviewer was recommended, including:
    - Keyword matches.
    - Similarity scores.
    - Past review performance metrics.

## **2.1.7 Reviewing Assigned Papers**

- **2.1.7.1 Visibility for Conference and Track Chairs**
  - Conference Chairs:
    - Can view all reviews submitted within their conference (e.g., RECOMB 2025).
  - Track Chairs:
    - Have access only to reviews specific to their assigned track (e.g., ISMB 2026 Sequence Analysis), maintaining confidentiality across tracks.
- **2.1.7.2 Personalized Reviewer Profiles**
  - Reviewers (PC members) can customize their profiles by specifying:
    - Keywords or topics they prefer to review.
    - Keywords or topics they wish to avoid.
    - Agreements to review specific tracks in shared multi-track conferences.

- **2.1.7.3 Conflict-of-Interest Detection**

- The system automates conflict-of-interest checks by cross-referencing:
  - Registered kinship statuses.
  - Past mentor-student relationships.
  - Institutional affiliations and recent collaborations.

- **2.1.7.4 Review Quality Ratings by Chairs**

- Track and conference chairs can rate reviews based on factors like clarity, depth, and helpfulness.
- These ratings help identify reliable reviewers for future conferences or tracks.
- Review ratings can optionally be shared across the same conference series to aid in future planning.
- Ratings remain anonymous to protect reviewer privacy and ensure fairness.
- Reviewers cannot access their own ratings or see feedback provided by chairs.
- Access to ratings is restricted to authorized users (e.g., chairs and organizers).

- **2.1.7.5 Peer Ratings for Reviewers**

- Reviewers can rate their peers' reviews on the same paper, focusing on:
  - Fairness of comments.
  - Constructiveness and relevance of feedback.
- Peer ratings contribute to maintaining accountability and enhancing review quality.

## **2.1.8 User Profile Page**

- **2.1.8.1 Profile Fields**

- The system provides each reviewer with a personalized profile page for managing their information and preferences.
- The profile includes the following fields:
  - Personal Information: Name and email address.
  - Professional Details: Current and past affiliations.
  - Expertise: Areas of expertise and research interests.

- Keywords: Topics they prefer to review or avoid.
- Conflict-of-Interest Declarations: Relationships or circumstances that may affect review impartiality.
- Track Preferences: Preferences for specific tracks in multi-track conferences.

- **2.1.8.2 Visibility for Organizers and Chairs**

- Certain fields, such as name, email, and areas of expertise, are optionally visible to conference organizers and track chairs to facilitate reviewer selection.
- The visibility is configurable, allowing reviewers to control what information is shared.

- **2.1.8.3 Privacy of Sensitive Details**

- Sensitive fields, such as conflicts of interest and personal preferences, remain private and are not accessible to conference organizers or chairs.
- These fields are used solely for automated checks and assignments.

- **2.1.8.4 Integration with External Platforms**

- Profile functionality mirrors established platforms like ORCID and Web of Science, making it familiar and user-friendly for reviewers.
- Reviewers can link their profiles to external accounts for streamlined management.

## **2.1.9 LLM-Generated Review Detection (Future Plan)**

- **2.1.9.1 Purpose of Detection Tools**

- The system will incorporate tools designed to identify and flag reviews generated by large language models (LLMs).
- This feature aims to maintain the authenticity and quality of submitted reviews, ensuring a fair review process.

- **2.1.9.2 Natural Language Processing (NLP) Techniques**

- Detection tools will leverage advanced NLP algorithms to analyze review content for patterns indicative of automation.
- Key analysis techniques may include:
  - Text Uniformity: Identifying overly generic or repetitive phrasing.
  - Lack of Contextual Depth: Detecting reviews that fail to engage

meaningfully with the paper's content.

- **Language Patterns:** Highlighting stylistic markers typical of LLM-generated text.

- **2.1.9.3 Improving Review Credibility**

- By identifying potentially automated reviews, the system ensures the credibility of the review process.
- This feature helps maintain trust among authors, reviewers, and organizers.

## **2.2 Non-functional Requirements**

- **2.2.1 Usability**

- The usability of the system will be driven by React, which will provide a dynamic, modern, and user-friendly interface for the platform. React enables the efficient development of a satisfying user experience with minimal load times and smooth interactions. Users will enjoy a consistent interface across different browsers, with features such as real-time feedback, easy navigation, and intuitive forms. React will also allow the front-end to integrate seamlessly with the back-end, ensuring that conference data is always up-to-date. Java Spring Boot will handle the back-end logic and data processing, enabling the front-end to fetch and display necessary information reliably.

- **2.2.2 Reliability**

- To ensure the reliability of the system, Java Spring Boot will handle the back-end services. The framework's built-in features, such as transaction management, error handling, and fault tolerance, will ensure that critical operations like paper submissions and reviewer assignments remain operational, even if some components experience failures. Spring Boot's integrated support for PostgreSQL through Hibernate will guarantee that the database is used efficiently, with automatic transaction management ensuring data integrity. Additionally, Spring Boot's monitoring and logging capabilities, specifically with the help of the actuator, will help the system proactively detect and recover from potential issues, ensuring the platform meets an uptime goal of 99.9%.

- **2.2.3 Performance**

- The performance of the system will be optimized through the combined use of React on the front-end and Java Spring Boot on the back-end. React will allow for fast, dynamic updates to the user interface, ensuring a responsive experience with sub-second load times for user actions such as filtering reviewer profiles or submitting papers. On the back-end, PostgreSQL will be used for efficient data storage, with Spring Boot's built-in support for optimizing database queries and

handling large volumes of data. The platform will also leverage asynchronous processing to perform time-consuming tasks in the background, ensuring that the user interface remains responsive during peak activity times, such as submission and review periods.

- **2.2.4 Scalability**

- While the system will use a monolithic architecture, it will be designed with scalability in mind to handle increasing traffic and growing amounts of conference data. React's modular structure will ensure that new features can be added without disrupting the user experience. On the back-end, Java Spring Boot will handle the scalability needs through its support for load balancing and horizontal scaling at the application level, without relying on microservices. PostgreSQL will be optimized to scale efficiently as the platform's data grows, ensuring that performance is maintained even as more conferences, submissions, and reviewers are added. By using Spring Boot's built-in tools for database access, such as Spring Data, and focusing on optimizing database queries, the platform will scale without a problem along with accommodating growing demand. The system's scalability aligns with the AWS Well-Architected Framework, ensuring that it can handle increased user demand while maintaining performance and reliability [1].

## **2.3 Pseudo Requirements**

1. The system supports multiple user roles.
2. Organizers can create and manage both single-track and multi-track conferences.
3. Tracks can have unique themes, submission criteria, and deadlines.
4. Organizers can send invitations to PC members individually or through bulk imports.
5. Reviewer-paper matching uses automated recommendations based on expertise and keywords.
6. Authors can submit their papers through a user-friendly interface.
7. Multiple file formats for submission are supported (e.g., PDF, Word.).
8. Authors can track submission statuses, from initial submission to final decision.
9. Organizers and reviewers can monitor review progress and completion.
10. The system automates compliance checks for formatting and file requirements.
11. Conflict-of-interest checks are conducted automatically based on affiliations and prior relationships.
12. PC members can declare conflicts of interest and update them as needed.

13. The system supports advanced workflows, including paper bidding and lead reviewer roles.
14. Organizers can define and assign roles (e.g., senior reviewer, track chair) for PC members.
15. Timeline dashboards display submission and review statuses for organizers and authors.
16. The system ensures fair workload distribution across reviewers in multi-track conferences.
17. Confidentiality between tracks is maintained for reviews and reviewer assignments.
18. Reviewer performance metrics, including timeliness and review quality, are tracked for future use.

## **3. Final Architecture and Design Details**

### **3.1 Software Architecture**

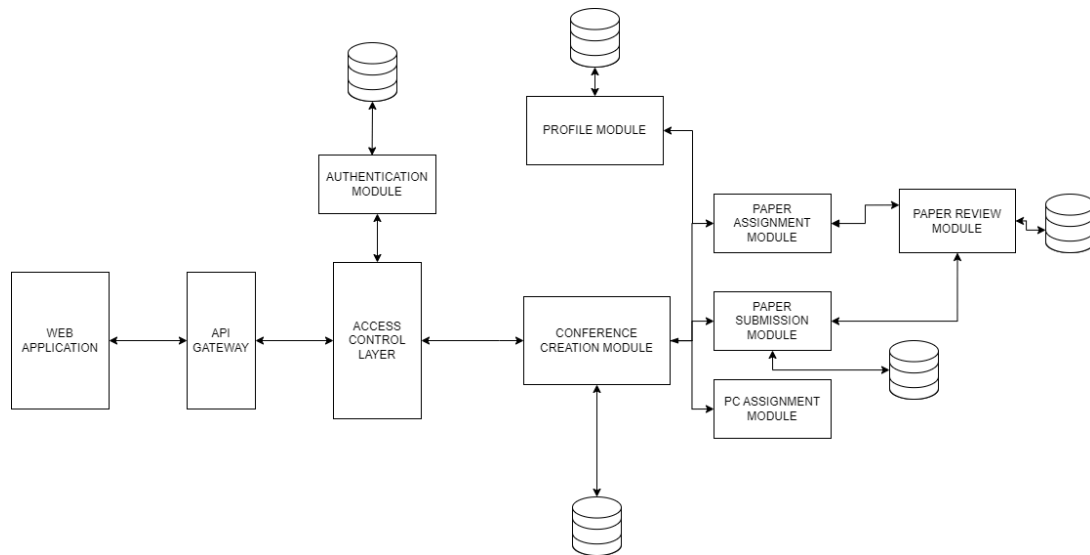
#### **3.1.1 Overview**

The system is structured to manage academic conferences efficiently by integrating various functional modules. The web application serves as the primary interface, communicating with backend services through an API gateway. Secure authentication and authorization are enforced by an access control layer. The system enables conference creation, paper submission, reviewer assignment, and peer review, with persistent data storage handled by dedicated databases.

#### **3.1.2 Subsystem Decomposition**

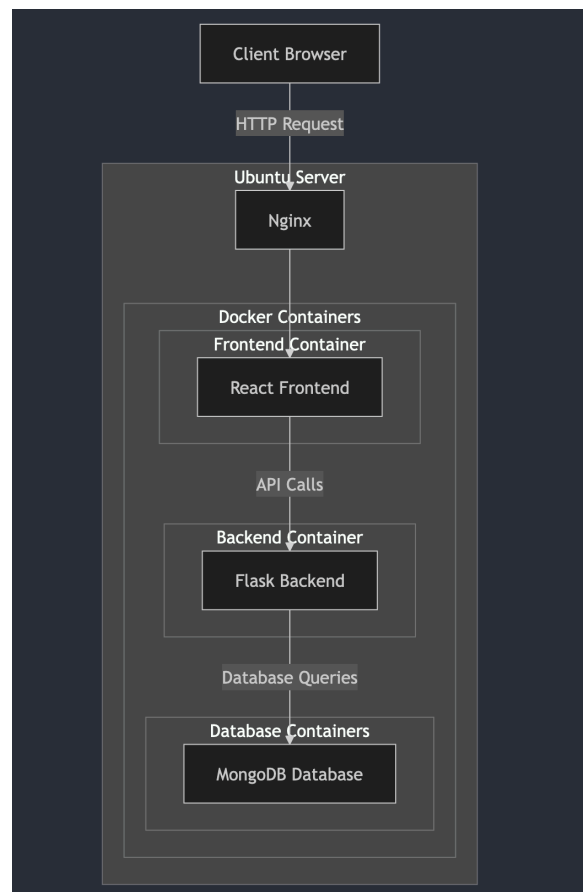
The system consists of multiple subsystems, each responsible for a specific function. The authentication module manages user access, while the conference creation module allows organizers to set up events. The paper submission module handles document uploads, and the PC assignment module designates program committee members. The paper assignment module distributes submissions to reviewers, and the paper review module facilitates the evaluation process. These subsystems interact through structured data flows, ensuring seamless conference management.





### 3.1.3 Hardware/software mapping

The system is deployed on an Ubuntu Server, which serves as the host machine running Docker containers for different software components. Nginx, installed on the host, acts as a reverse proxy, forwarding client requests to the appropriate services. The React Frontend runs inside a dedicated Docker container, handling user interactions and communicating with the backend via API calls. The Flask Backend is also containerized, processing business logic and interacting with the MongoDB Database, which resides in its own container for data storage and retrieval. The system runs on a physical or virtual machine with sufficient CPU, memory, and storage to support containerized workloads efficiently.



### 3.1.4 Persistent data management

The system uses MongoDB as its primary database for storing and managing conference-related data, including user profiles, submissions, reviews, and assignments. Each module interacts with MongoDB to store structured documents, ensuring flexibility in handling different data formats. Paper submissions and reviews are linked through document references, enabling efficient querying and retrieval. Data consistency is maintained by atomic operations and appropriate indexing strategies to optimize performance.

### 3.1.5 Access control and security

Our system has many user types with very specific roles, which must be handled very accurately in order to function as a proper conference organization system. Thus, the system has emphasized on the design of access control and security. It enforces access control and security through a dedicated authentication module, implemented using Flask and integrated with MongoDB. User authentication is handled via token-based mechanisms such as JWT, ensuring secure session management. Role-based access control (RBAC) restricts actions based on user roles, preventing unauthorized access. Sensitive data is encrypted, and secure API endpoints are enforced using HTTPS, protecting communication between clients and backend services.

## 3.2 Subsystem Services

The Conferencer system is decomposed into several core subsystems, each responsible for a specific aspect of conference management. This modular structure enforces separation of concerns, fosters scalability, and simplifies maintenance.

### 3.2.1 Authentication and Access Control Service

Manages user authentication, authorization, and session handling across the Conferencer platform.

- **User Login/Logout:** Verifies credentials (email/password, ORCID, or Google) and issues session tokens (e.g., JWT).
- **Role-Based Access Control:** Determines if a user (Superchair, Track Chair, PC Member, Reviewer, Author) has access to specific functionalities.
- **Session Management:** Tracks active sessions, handles timeouts, and enforces auto-logout policies.
- **Conflict-of-Interest Checks:** Cross-references user affiliations for basic conflict detection prior to PC assignments.

### 3.2.2 Conference Management Service

Enables Superchairs (and authorized users) to create new conferences or import data from previous events.

- **New Conference Setup:** Collects conference details (name, acronym, submission deadlines, tracks).
- **Import & Configuration:** Loads settings/members from older conferences or Excel files.
- **Conference Overview:** Displays summary of participants, tracks, and global settings.
- **Policy Enforcement:** Applies double-blind or single-blind review rules as configured.

### 3.2.3 Submission Management Service

Provides authors with an interface to submit their manuscripts and relevant metadata to specific tracks.

- **File Upload & Metadata:** Supports PDF (or other permissible formats) with title, abstract, keywords.
- **Submission Tracking:** Allows authors to view the status of each submission (in review, accepted, or rejected).
- **Automated Compliance Checks:** Validates page limits, file size, and required fields.
- **Revision Handling:** Permits re-uploads before the submission deadline.

### 3.2.4 Review Assignment and Management Service

Manages roles and memberships within a conference's Program Committee, including invitations and conflict-of-interest declarations.

**PC Invitations:** Sends invitations in bulk or individually (via email or integrated platforms) to prospective committee members.

- **Role Management:** Grants roles like Superchair, Track Chair, or PC Member, and keeps track of each user's permissions.
- **Conflict-of-Interest Handling:** Automates detection of potential conflicts based on user affiliations, co-authorships.
- **Historical Performance:** Tracks PC members' past reviews, timeliness, and overall ratings.

### 3.2.5 Decision-Making Service

Assigns submissions to the right reviewers, ensuring fair distribution of workload and alignment with expertise.

- **Reviewer Matching:** Suggests optimal reviewers based on track, keywords, expertise, and conflict-of-interest data.
- **Load Balancing:** Prevents excessive assignment of papers to any single reviewer.
- **Manual Adjustments:** Allows chairs to override automated recommendations for special cases.
- **Multi-Track Confidentiality:** Ensures track chairs only access papers/reviews within their assigned tracks.

### 3.2.6 User Profile and Analytics Service

Manages the entire review workflow: from initial reviewer feedback to final accept/reject decisions. **Review Submission:** Collects scores, comments, and attached critique documents from each reviewer.

- **Peer Rating:** Optionally allows reviewers to rate the quality of each other's reviews.
- **Chair Decision:** Supports track chairs in finalizing accept/reject decisions, referencing aggregated reviewer feedback.
- **LLM-Generated Review Detection (Future):** Plans to integrate an NLP-based mechanism for detecting AI-generated reviews.

### 3.2.7 Notification and Communication Service

Handles user profiles, including personal data, historical roles, and review performance metrics.

**Profile Editing:** Allows users to update affiliations, bios, and conflict-of-interest declarations.

- **Roles & Permissions:** Displays assigned roles (reviewer, track chair, etc.) and relevant privileges.
- **Performance Statistics:** Tracks average review time, number of accepted/rejected papers, and rating from chairs.
- **Integration:** Links with external platforms (e.g., ORCID, Web of Science) for verified user profiles.

## 4. Development/Implementation Details

The development and implementation of *Conferencer* resulted in a scalable, secure, and user-friendly web application supporting academic conference workflows, integrating user authentication, conference and track management, submissions, reviewing, reviewer assignment, and notifications. The following subsections detail the technologies used, frontend and backend, database design, security measures, and deployment approach.

### 4.1 Technology Stack

Component	Technology
Frontend	React (JavaScript, Context API, React Router)
Backend	Flask (Python), Flask-RESTful
Database	MongoDB
Authentication	Session-based authentication (updated from JWT), OAuth 2.0 integrations (Google, ORCID)
Containerization	Docker
Web Server / Reverse Proxy	Nginx
Deployment	Ubuntu VPS
Development Tools	Visual Studio Code, Git, GitHub, Jira

This stack was selected to balance development speed with long-term maintainability, while also offering robust support for security and scalability.

## 4.2 Frontend Implementation

The **React** frontend was developed to provide a responsive, intuitive, and accessible user interface. The UI was modularized into pages and components to ensure reusability and maintainability.

Key user-facing features included:

- **Authentication Interfaces:** User login and registration with session-based authentication and OAuth options for Google and ORCID.
- **Conference and Track Management:** Interfaces for creating and configuring conferences and their tracks.
- **Paper Submission Workflow:** Forms and dashboards for uploading, editing, and managing paper submissions.
- **Reviewer Assignment Panels:** Tools for assigning reviewers to submissions, including filtering options.
- **Review Management:** Interfaces for submitting and managing reviews, with review ratings and reviewer history.
- **Notification Center:** Displaying system-generated notifications, including assignment updates and review submissions.
- **User Profile Management:** Editable profiles supporting personal information updates and conflict declarations.
- **Task Management:** My Tasks dashboard showing pending actions for each user role.
- **Reviewer Statistics:** Dedicated statistics page displaying reviewer performance metrics and review quality ratings.
- **Chat:** A chat system enabling communication between reviewers and conference organizers.
- **Search and Filtering:** Tools for filtering submissions and reviews based on user-defined criteria.
- **Responsive Design:** The interface dynamically adapts to different screen sizes and devices.

State management was handled using the Context API and React Hooks to enable smooth data flow and efficient state updates across components. Usability feedback gathered during development was incorporated into ongoing refinements.

### 4.3 Backend Implementation

The **Flask** backend provided the business logic, data processing, and API endpoints that supported all system operations.

Key services included:

- **Authentication and Authorization:** Secure session-based login/logout, OAuth integration, and strict role-based access control (RBAC).
- **Conference and Track Management:** Handling creation, updating, and retrieval of conference and track data.
- **Submission Service:** Managing paper submissions, metadata, and version control.
- **Reviewer Assignment Service:** Facilitating manual reviewer-paper matching and storing assignment data.
- **Review Management:** Enabling review submissions and updates, and storing review data.
- **Notification Service:** Generating and delivering notifications based on user actions and system events.
- **Profile Management:** Supporting profile editing, conflict declarations, and role updates.
- **Task Management:** Assigning and tracking user tasks across different roles.
- **Statistics Service:** Calculating and presenting reviewer performance metrics and review quality ratings.
- **Communication:** Supporting chat functionality between reviewers and conference organizers.

The backend architecture emphasized modularity, scalability, and maintainability, with clear separation of concerns among services. All services included thorough input validation, error handling, and data consistency mechanisms.

### 4.4 Database Design

The **MongoDB** database stored all persistent data and was designed to support flexibility, scalability, and efficient querying.

Primary data entities included:

- **Users:** Storing user accounts, authentication data, roles, and profile details.
- **Conferences and Tracks:** Capturing conference configurations and associated tracks.
- **Submissions:** Managing uploaded papers, metadata, and version histories.
- **Reviewer Assignments:** Recording reviewer-paper relationships.
- **Reviews:** Storing submitted reviews, scores, and reviewer history.
- **Notifications:** Logging notifications and their delivery statuses.
- **Keywords:** Maintaining keyword data for filtering and recommendation purposes.
- **Program Committee Invitations:** Recording invitations and statuses for program committee members.
- **Statistics:** Storing reviewer performance metrics and review quality data.
- **Chat Messages:** Storing communication logs for the chat feature.

Indexing was applied to commonly queried fields to optimize performance, particularly for user searches, paper filtering, and reviewer assignment operations.

## 4.5 Access Control and Security

Security was integrated into every aspect of the system:

- **Session-Based Authentication:** Providing secure and efficient session management.
- **OAuth 2.0 Integration:** Enabling secure third-party login via Google and ORCID.
- **Role-Based Access Control (RBAC):** Restricting access to actions based on user roles.
- **Input Validation and Sanitization:** Protecting against injection attacks, cross-site scripting (XSS), and similar vulnerabilities.
- **Secure File Uploads:** Validating and securely storing uploaded papers.
- **Audit Logging:** Tracking key user actions for accountability and traceability.
- **Testing and Validation:** Conducting security tests to verify that unauthorized actions were blocked and access restrictions functioned correctly.

## 4.6 CI/CD and Deployment

The system was deployed using a combination of containerization and manual continuous integration and deployment (CI/CD) processes:

- **Version Control:** Managed with Git and GitHub, following a feature branch



workflow with pull request reviews.

- **Containerization:** Frontend, backend, and database services were containerized using Docker, enabling consistent development and production environments.
- **Web Server Configuration:** Nginx served the React frontend and routed API requests to the Flask backend.
- **Deployment Environment:** The system was hosted on a Virtual Private Server (VPS) running Ubuntu.
- **Deployment Process:** Manual CI/CD practices were followed, including feature testing and validation before each deployment.

This approach enabled efficient development cycles, minimized downtime, and ensured consistent system performance across development and production environments.

## 5. Testing Detail

### Non Functional Test Cases

Test ID	NFT-PERFORMANCE-001
Test Type	Performance Test
Test Objective	To assess the system's response time and performance under a high load of simultaneous real-time chat users.
Procedures	<ul style="list-style-type: none"><li>● Simulate multiple concurrent users sending messages in the real-time chat module.</li><li>● Measure the message delivery time for each user.</li><li>● Monitor server CPU, memory, and bandwidth usage during the test.</li><li>● Identify if any delays or timeouts occur.</li></ul>
Expected Results	<ul style="list-style-type: none"><li>● Message delivery time should not exceed 2 seconds.</li><li>● No significant server resource bottlenecks.</li><li>● No message loss or chat disruptions.</li></ul>
Test Priority	Critical
Date Tested	28/04/2025
Results	Passed

Test ID	NFT-SECURITY-001
---------	------------------

Test Type	Security Test
Test Objective	To verify the security of the chat and notification systems against unauthorized access.
Procedures	<ul style="list-style-type: none"> <li>● Attempt to access chat messages of other users without authorization.</li> <li>● Perform SQL injection tests on chat input fields.</li> <li>● Test for XSS vulnerabilities in the forum and notification messages.</li> <li>● Verify that sensitive data (e.g., user tokens) is encrypted during transmission.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>● Unauthorized access should be denied with appropriate error messages.</li> <li>● SQL injection and XSS attempts should fail.</li> <li>● All sensitive data should be encrypted.</li> </ul>
Test Priority	Critical
Date Tested	25/04/2025
Results	Passed

Test ID	NFT-USABILITY-001
Test Type	Usability Test
Test Objective	To assess the ease of use of the notification and chat systems for new users.
Procedures	<ul style="list-style-type: none"> <li>● Recruit a group of new users unfamiliar with the system.</li> <li>● Ask them to access and respond to notifications and send messages via chat.</li> <li>● Collect feedback on ease of use, clarity of icons, and intuitiveness of navigation.</li> <li>● Record the time taken to complete tasks.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>● Users should be able to perform tasks without guidance.</li> <li>● Navigation should be intuitive, and icons should be self-explanatory.</li> <li>● Average task completion time should be under 2 minutes.</li> </ul>

Test Priority	Minor
Date Tested	01/05/2025
Results	Passed

Test ID	NFT-COMPATIBILITY-001
Test Type	Compatibility Test
Test Objective	To verify that chat, notifications, and forums work across different browsers and devices.
Procedures	<ul style="list-style-type: none"> <li>• Test the system on Chrome, Firefox, Safari, and Edge.</li> <li>• Test on desktop (Windows, macOS) and mobile devices (iOS, Android).</li> <li>• Verify that messages, notifications, and forum posts appear correctly on all platforms.</li> <li>• Check if file attachments can be sent and opened on different devices.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>• Consistent appearance and functionality across all tested browsers and devices.</li> <li>• No UI/UX distortions or functional errors.</li> </ul>
Test Priority	Critical
Date Tested	28/04/2025
Results	Passed

Test ID	NFT-RELIABILITY-001
Test Type	Reliability Test
Test Objective	To assess the reliability of chat and notification services over extended use.
Procedures	<ul style="list-style-type: none"> <li>• Run a continuous chat test for 24 hours, sending messages every 5 seconds.</li> <li>• Restart the server multiple times and verify that chat history and notifications are retained.</li> <li>• Monitor system logs for errors or crashes.</li> </ul>

Expected Results	<ul style="list-style-type: none"> <li>• No system crashes or memory leaks.</li> <li>• All messages and notifications should persist correctly after the server restarts.</li> </ul>
Test Priority	Critical
Date Tested	28/04/2025
Results	Passed

Test ID	NFT-SCALABILITY-001
Test Type	Scalability Test
Test Objective	To evaluate how the chat and notification systems handle increasing numbers of users.
Procedures	<ul style="list-style-type: none"> <li>• Simulate a gradual increase in users from 100 to 10,000.</li> <li>• Measure system response time and resource utilization at each step.</li> <li>• Test if notifications and chat messages are still delivered promptly.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>• System should scale without significant increases in response time.</li> <li>• Resource utilization should remain within acceptable limits.</li> </ul>
Test Priority	Critical
Date Tested	
Results	N/A

Test ID	NFT-SUBMISSION-005
Test Type	Usability Test
Test Objective	Verify file uploads can handle specific filetypes.
Procedures	<ul style="list-style-type: none"> <li>• Check if file upload interface only lists .pdf files.</li> <li>• Check these pdf files get stored in the database, and can be downloaded after submissions by clicking on them in “My Submissions list.”</li> </ul>

Expected Results	<ul style="list-style-type: none"> <li>Only .pdf files are allowed to be uploaded.</li> <li>These files correctly download after clicking on them in the list.</li> </ul>
Test Priority	Minor
Date Tested	26/04/2025
Results	Passed

Test ID	NFT-NOTIFICATION_DELIVERY-001
Test Type	Reliability Test
Test Objective	Ensure notifications (email & in-app) deliver reliably without delays.
Procedures	<ul style="list-style-type: none"> <li>Assign a reviewer to a paper.</li> <li>Verify that an in-app notification appears in the notification center.</li> <li>Check if an email notification is received within 1-2 minutes.</li> <li>Log in as the reviewer and click on the notification.</li> <li>Ensure that it redirects to the assigned paper review page.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>Notifications should deliver instantly.</li> <li>Clicking a notification should navigate to the correct page.</li> </ul>
Test Priority	Major
Date Tested	29/04/2025
Results	Partially Passed*

\*: As of the time of the testing for the final report, mail function is not implemented yet. However other notification mediums did pass the test.

Test ID	NFT-SESSION_HANDLING-001
Test Type	Security Test
Test Objective	Ensure that inactive user sessions expire after a defined timeout period.
Procedures	<ul style="list-style-type: none"> <li>Log in to the system and stay inactive for 15+ minutes.</li> <li>Try accessing the dashboard after inactivity.</li> <li>Verify if the system logs the user out automatically.</li> <li>Try performing an action (e.g., submitting a review) after timeout.</li> </ul>

Expected Results	<ul style="list-style-type: none"> <li>● Inactive users should be automatically logged out.</li> <li>● Any session-based actions should redirect to the login page.</li> </ul>
Test Priority	Critical
Date Tested	22/04/2025
Results	Passed

Test ID	NFT-BROWSER_COMPATIBILITY-001
Test Type	Compatibility Test
Test Objective	Verify that the system works on different browsers.
Procedures	<ul style="list-style-type: none"> <li>● Open the platform on Google Chrome, Mozilla Firefox, Microsoft Edge.</li> <li>● Log in and perform key actions (paper submission, review, role switching).</li> <li>● Check for UI inconsistencies or missing functionalities.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>● The application should be fully functional across all major browsers.</li> <li>● No UI components should break.</li> </ul>
Test Priority	Major
Date Tested	
Results	N/A

### 1.1. Functional Test Cases

Test ID	FT-SUBMISSION-001
Test Type	Functional Test
Test Objective	Verify paper submission for authors.
Procedures	<ul style="list-style-type: none"> <li>● In submissions screen, click on “Add Submission” button</li> <li>● Check if “Add Submission” page shows up with relevant data input fields.</li> <li>● Check if track selections can be changed, if multiple tracks exist.</li> <li>● Check if the text fields can be filled with inputs, and the file can be uploaded to the system</li> <li>● Check if upon completion, a confirmation display with the</li> </ul>

	correct information shows up and the “My Submissions” page displays the added submission with the correct information.
Expected Results	<ul style="list-style-type: none"> <li>• After clicking on the button, “Add Submissions” page opens up.</li> <li>• Track selections change after clicking on the radio buttons.</li> <li>• Input fields correctly display the entered inputs.</li> <li>• After completion, a confirmation screen opens up with the information displayed correctly.</li> <li>• “My Submissions” page displays the added submission correctly</li> </ul>
Test Priority	Critical
Date Tested	28/04/2025
Results	Passed

Test ID	FT-SUBMISSION-002
Test Type	Functional Test
Test Objective	Verify paper submissions can be edited after the initial submission.
Procedures	<ul style="list-style-type: none"> <li>• In submissions screen, click on “Edit Paper” button</li> <li>• Check if a “Edit Paper” page shows up with the relevant input fields already selected/filled with information</li> <li>• Check if track selections can be changed, the text fields can be changed with different inputs, and a different file can be reuploaded to the system</li> <li>• Check if upon completion, a confirmation display with the correct information shows up and “My Submissions” page displays the edited submission with the correct information.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>• After clicking on the button, “Edit Paper” page opens up with the previously entered paper details correctly filled in the fields.</li> <li>• Track selections change after clicking on the radio buttons. Inputs can be correctly edited.</li> <li>• After completion, a confirmation screen opens up with the information displayed correctly.</li> <li>• “My Submissions” page displays the edited submission correctly</li> </ul>
Test Priority	Major
Date Tested	28/04/2025
Results	Passed

Test ID	FT-SUBMISSION-003
Test Type	Functional Test
Test Objective	Verify paper submissions can be deleted after the initial submission.
Procedures	<ul style="list-style-type: none"> <li>• In submissions screen, click on “Delete Paper” button</li> <li>• Check if a confirmation pop up shows up with the information about the results of the action</li> <li>• Check if upon clicking “Confirm” button, a confirmation display shows up, and “My Submissions” page does not display the deleted submission any longer.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>• After clicking on the button, a confirmation pop up shows up.</li> <li>• After confirming, the review is deleted on “My Submissions” list.</li> </ul>
Test Priority	Minor
Date Tested	28/04/2025
Results	Passed

Test ID	IT-SUBMISSION-004
Test Type	Integration Test
Test Objective	Verify paper submissions’ results can be tracked.
Procedures	<ul style="list-style-type: none"> <li>• Paper review process finalizes and a decision is made by the authorized person regarding the paper.</li> <li>• In “My Submissions” screen, check if this final decision is displayed to the author.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>• “My Submissions” lists the results of papers with finalized decisions correctly.</li> </ul>
Test Priority	Major
Date Tested	28/04/2025
Results	Passed

Test ID	UT-TASKS-001
Test Type	Usability Test
Test Objective	Verify that reviewers can organize their tasks by sorting and filtering.
Procedures	<ul style="list-style-type: none"> <li>• In “My Tasks” screen click on “Sort” button and select which</li> </ul>



	<p>direction and what attribute to sort the papers by.</p> <ul style="list-style-type: none"> <li>● Check if sorting is done correctly.</li> <li>● In “My Tasks” screen click on “Filter” button and select what attributes to filter the papers by.</li> <li>● Check if filtering is done correctly.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>● After clicking on the “Sort” button, and the selections are made, awaiting papers are sorted correctly by the given definitions.</li> <li>● After clicking on the “Filter” button, and the selections are made, awaiting papers are sorted correctly by the given definitions.</li> </ul>
Test Priority	Low
Date Tested	
Results	

Test ID	FT-TASKS-002
Test Type	Functional Test
Test Objective	Verify reviewers can download the papers they were assigned to in a quick and correct manner.
Procedures	<ul style="list-style-type: none"> <li>● In “My Tasks” screen, check if already assigned and unreviewed papers are listed correctly.</li> <li>● Click on the paper .pdf file and check if download starts within 5 seconds.</li> <li>● Check if the downloaded paper matches the file of the paper clicked on.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>● My Tasks screen lists the reviewed/unreviewed papers correctly.</li> <li>● Download starts within 5 seconds and the file matches the correct paper.</li> </ul>
Test Priority	Critical
Date Tested	28/04/2025
Results	Passed

Test ID	FT-TASKS-003
Test Type	Functional Test
Test Objective	Verify reviewers can view the papers they were assigned to.

Procedures	<ul style="list-style-type: none"> <li>● In “My Tasks” screen, check if already assigned and unreviewed papers are listed correctly.</li> <li>● After a new assignment is made by the authorized users to the reviewer, new assignment shows up in the list.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>● My Tasks screen lists the reviewed/unreviewed papers correctly.</li> <li>● When a new entry is made, it is also displayed correctly after refreshing of the page.</li> </ul>
Test Priority	Critical
Date Tested	28/04/2025
Results	Passed

Test ID	IT-TASKS-004
Test Type	Integration Test
Test Objective	Verify that tasks list update after an edit or deletion to the relevant papers.
Procedures	<ul style="list-style-type: none"> <li>● After a delete or edit is done by the author to a relevant paper, in “My Tasks” screen, check if this paper is updated with the correct information or deleted entirely.</li> <li>● Check if an information display is shown regarding the update.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>● The relevant paper is updated with the correct information or deleted entirely, after the first refresh of the page.</li> <li>● An information display is shown with the regarding update.</li> </ul>
Test Priority	Minor
Date Tested	28/04/2025
Results	Passed

Test ID	FT-TASKS-005
Test Type	Functional Test
Test Objective	Verify reviewers can upload reviews to the papers they were assigned to.
Procedures	<ul style="list-style-type: none"> <li>● In “My Tasks” screen, after the “Make Review” button is clicked, check if “Make Review” page opens up.</li> <li>● Check if this page displays the correct paper, with the correct information.</li> </ul>

	<ul style="list-style-type: none"> <li>● Check if review text can be uploaded, or typed on the textfield. Also, check if decision and confidence score can be selected by the given ranges by the conference authorities.</li> <li>● Check if these required fields are filled before completion.</li> <li>● Upon completion, check if a confirmation page shows up, and tasks are updated accordingly to the new review</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>● After “Make Review” button is clicked, check if “Make Review” page opens up</li> <li>● After the inputs have been filled, and the complete button clicked, the inputs are validated. If successful, taken to a confirmation display with the correct information.,</li> <li>● My Tasks list is updated with the new review being listed as done, and the paper is taken off of the “Unreviewed Papers” list.</li> </ul>
Test Priority	Critical
Date Tested	29/04/2025
Results	Passed

Test ID	FT-TASKS-006
Test Type	Functional Test
Test Objective	Verify reviewers can edit and delete their submitted reviews after submission.
Procedures	<ul style="list-style-type: none"> <li>● In the “Reviewed Papers” screen, click on the “Update Review” button.</li> <li>● Check if the “Edit Review” page opens up with the previously entered review details correctly filled in the fields.</li> <li>● Verify that the review text, decision, and confidence score can be modified.</li> <li>● Check if, after submitting the changes, a confirmation message appears, and the updated review is displayed in the “Reviewed Papers” list with the correct information.</li> <li>● In the “Reviewed Papers” screen, click on the “Delete Review” button.</li> <li>● Check if a confirmation pop-up appears with details about the deletion.</li> <li>● Verify that upon clicking the “Confirm” button, the review is removed from the “Reviewed Papers” list.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>● After clicking on the “Update Review” button, the “Edit Review” page opens with the previous review details correctly filled.</li> <li>● The review text, decision, and confidence score can be edited, and upon submission, a confirmation display appears with the</li> </ul>

	<p>updated review information.</p> <ul style="list-style-type: none"> <li>• After clicking the “Delete Review” button, a confirmation pop-up appears.</li> <li>• Upon confirming, the review is deleted and no longer appears in the “Reviewed Papers” list.</li> </ul>
Test Priority	Major
Date Tested	29/04/2025
Results	Passed

Test ID	IT-PROFILE-001
Test Type	Integration Test
Test Objective	Verify that the statistics on the profile page update correctly when a new review is uploaded.
Procedures	<ul style="list-style-type: none"> <li>• Navigate to the profile page and record the current statistics displayed.</li> <li>• Upload a new review for an assigned paper.</li> <li>• After submission, navigate back to the profile page and check if the relevant statistics, such as "Average Time To Review," "Average Submit Time Before Deadline," "Deadline Compliance Rate," "Average Rating Given," "Average Words Per Review," and "Review Rating," are updated accordingly.</li> <li>• Refresh the page and confirm that the updated values persist.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>• After a new review is uploaded, the profile statistics update to reflect the new review data.</li> <li>• The updated statistics remain correct and persist after refreshing the page.</li> </ul>
Test Priority	Major
Date Tested	29/04/2025
Results	Passed

Test ID	IT-PROFILE-002
Test Type	Integration Test
Test Objective	Verify that the active and past roles on the profile page updated correctly when a new review is uploaded.
Procedures	<ul style="list-style-type: none"> <li>• Navigate to the profile page and check if active and past roles are correctly displayed under the respective sections.</li> </ul>

	<ul style="list-style-type: none"> <li>● For each active role, click on the symbol to access the corresponding conference page.</li> <li>● If the user has authorization, verify that the conference page opens successfully.</li> <li>● If the user does not have authorization, check if an error message pop-up appears stating that access is denied.</li> <li>● Have an authorized user assign a new role to the current user.</li> <li>● Refresh the profile page and verify that the new role appears under "Active Roles."</li> <li>● If an existing role expires or is removed, check if it moves to "Past Roles" accordingly.</li> <li>● Verify that access permissions to the newly assigned role's conference page function correctly based on authorization.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>● Active and past roles are correctly categorized and displayed.</li> <li>● Clicking on the conference access symbol grants access to the page if the user has the appropriate authorization.</li> <li>● If unauthorized, an error message pop-up is displayed, preventing access.</li> <li>● When a new role is assigned, it appears under "Active Roles" after refreshing.</li> <li>● If a role expires or is removed, it moves to "Past Roles."</li> <li>● Role information and access permissions remain consistent after refreshing the page</li> </ul>
Test Priority	Major
Date Tested	29/04/2025
Results	Passed

Test ID	UT-CONFERENCE-001
Test Type	Usability Test
Test Objective	Verify that the filtering functionality correctly filters submissions based on selected tracks and keywords.
Procedures	<ul style="list-style-type: none"> <li>● Navigate to the conference submissions page.</li> <li>● Click on the "Filter" button to open the filtering options.</li> <li>● Select a track (e.g., "TRACK1") from the "Filter By Track" section.</li> <li>● Enter keywords (e.g., "machine learning") in the "Filter By Keywords" section.</li> <li>● Click the "Filter" button.</li> </ul>

Expected Results	<ul style="list-style-type: none"> <li>• The displayed list of submissions updates to show only those belonging to the selected track.</li> <li>• Only submissions containing the specified keywords in their metadata (title, abstract, or keywords) are shown.</li> <li>• The filtering options remain visible for further adjustments.</li> <li>• If no submissions match the filters, an appropriate message is displayed.</li> </ul>
Test Priority	Major
Date Tested	28/04/2025
Results	Passed

Test ID	UT-CONFERENCE-002
Test Type	Usability Test
Test Objective	Verify that the search functionality correctly filters submissions based on the entered title or author name.
Procedures	<ul style="list-style-type: none"> <li>• Navigate to the submissions page.</li> <li>• Click on the search bar.</li> <li>• Enter a title or author name (e.g., “Virtual Reality in Education” or “Jane Doe”).</li> <li>• Press the “Enter” key or click on the search icon.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>• The displayed list of submissions updates to show only those that match the entered title or author name.</li> <li>• If multiple results match, all relevant submissions are displayed.</li> <li>• If no submissions match the search criteria, an appropriate message is displayed.</li> <li>• The search query remains in the search bar for user reference.</li> </ul>
Test Priority	Critical
Date Tested	27/04/2025
Results	Passed

Test ID	FT-CONFERENCE-003
---------	-------------------

Test Type	Functional Test
Test Objective	Verify that clicking the “View Reviews” button opens the review details for the corresponding paper.
Procedures	<ul style="list-style-type: none"> <li>• Navigate to the conference “Reviews” page.</li> <li>• Identify a paper with a “View Reviews” button.</li> <li>• Click the “View Reviews” button for that paper.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>• The system loads the review details page for the selected paper.</li> <li>• The page displays the assigned and completed reviews for that paper.</li> <li>• The user can navigate back to the reviews list if needed.</li> </ul>
Test Priority	Major
Date Tested	29/04/2025
Results	Passed

Test ID	IT-CONFERENCE-004
Test Type	Integration Test
Test Objective	Ensure that the displayed review completion status is accurate for each paper.
Procedures	<ul style="list-style-type: none"> <li>• Navigate to the conference “Reviews” page.</li> <li>• Observe the “Completed/Assigned Reviews” status for multiple papers.</li> <li>• Compare the displayed status with the actual number of completed reviews stored in the system.</li> <li>• Make updates such that the status of the papers change, verify if these changes show up on the page.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>• The “Completed/Assigned Reviews” count matches the actual data in the system.</li> <li>• If a reviewer completes a review, the count updates correctly upon page refresh.</li> </ul>

	<ul style="list-style-type: none"> <li>• No incorrect counts or inconsistencies appear across different user roles.</li> <li>• Updates coming from different users are displayed correctly upon refresh of the page.</li> </ul>
Test Priority	Low
Date Tested	29/04/2025
Results	Passed

Test ID	FT-TRACK_CHAIR-001
Test Type	Functional Test
Test Objective	Verify that the track chair can edit the decision for a paper in the review section.
Procedures	<ul style="list-style-type: none"> <li>• Log in as the track chair.</li> <li>• Navigate to a paper.</li> <li>• Click on the "Edit Decision" button.</li> <li>• Select a new decision from the dropdown (e.g., "Accept").</li> <li>• Save the changes.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>• The decision for the paper is updated to "Accept".</li> <li>• The updated decision is reflected in the "Current decision" field.</li> </ul>
Test Priority	Major
Date Tested	30/04/2025
Results	Passed

Test ID	FT-TRACK_CHAIR-002
Test Type	Functional Test
Test Objective	Verify that the track chair can assign an additional reviewer to the paper.
Procedures	<ul style="list-style-type: none"> <li>• Log in to the system with track chair credentials.</li> <li>• Navigate to the "CONFERENCE" section under "MY TASKS".</li> <li>• Locate the paper.</li> <li>• Click on the "Assign More Reviewers" button.</li> <li>• Select a reviewer from the list (e.g., "Napoleon") who is</li> </ul>



	<p>currently marked as "Pending".</p> <ul style="list-style-type: none"> <li>Confirm the assignment by clicking "Assign".</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>The reviewer "Napoleon" is added to the list of reviewers for the paper.</li> <li>The "Pending" status for "Napoleon" is updated with a deadline and review details.</li> <li>The reviewer's profile link remains accessible.</li> </ul>
Test Priority	Critical
Date Tested	29/04/2025
Results	Passed

Test ID	FT-TRACK_CHAIR-003
Test Type	Functional Test
Test Objective	Verify that the track chair can edit the decision for the paper in the "CONFERENCE" section.
Procedures	<ul style="list-style-type: none"> <li>Log in to the system with track chair credentials.</li> <li>Navigate to the "CONFERENCE" section under "MY TASKS".</li> <li>Locate the paper.</li> <li>Click on the "Edit Decision" button next to the "Current decision: None" field.</li> <li>Select a new decision (e.g., "Accept") from the dropdown menu.</li> <li>Click "Save" to confirm the changes.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>The "Current decision" field updates to "Accept".</li> <li>The decision change is reflected in the paper's review status.</li> </ul>
Test Priority	Major
Date Tested	29/04/2025
Results	Passed

Test ID	FT-TRACK_CHAIR-004
Test Type	Functional Test
Test Objective	Verify that the track chair can add a decision (Accept or Reject) for a

	paper in the "Add Decision For Paper" interface.
Procedures	<ul style="list-style-type: none"> <li>● Log in to the system with track chair credentials.</li> <li>● Navigate to the paper review section where the "Add Decision For Paper" interface is displayed.</li> <li>● Select either "Accept" or "Reject" from the available options.</li> <li>● Click on the "ADD DECISION" button to confirm the selection.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>● The selected decision ("Accept" or "Reject") is successfully applied to the paper.</li> <li>● The decision is reflected in the paper's review status.</li> <li>● The interface confirms the decision has been saved (e.g., via a success message or status update).</li> </ul>
Test Priority	Critical
Date Tested	27/04/2025
Results	Passed

Test ID	FT-TRACK_CHAIR-005
Test Type	Functional Test
Test Objective	Verify that the track chair can edit the decision for the paper in the "CONFERENCE" section.
Procedures	<ul style="list-style-type: none"> <li>● Log in to the system with track chair credentials.</li> <li>● Navigate to the "CONFERENCE" section under "MY TASKS".</li> <li>● Locate the paper.</li> <li>● Click on the "Edit Decision" button next to the "Current decision" field.</li> <li>● Select a new decision (e.g., "Accept") from the available options.</li> <li>● Click "Save" to confirm the changes.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>● The "Current decision" field updates to "Accept".</li> <li>● The decision change is reflected in the paper's review status.</li> </ul>
Test Priority	Major
Date Tested	27/04/2025
Results	Passed

Test ID	UT-TRACK_CHAIR-006
Test Type	Usability Test
Test Objective	Verify that the track chair can switch back to the reviews section from the "CREATE CONFERENCE" or "CHANGE ROLE" interface.
Procedures	<ul style="list-style-type: none"> <li>● Log in to the system with track chair credentials.</li> <li>● Navigate to the "CREATE CONFERENCE" or "CHANGE ROLE" section.</li> <li>● Click on the "Switch Back to Reviews" button.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>● The interface redirects to the reviews section.</li> <li>● The list of papers and their review statuses are displayed correctly.</li> </ul>
Test Priority	Major
Date Tested	28/04/2025
Results	Passed

Test ID	FT-TRACK_CHAIR-007
Test Type	Functional Test
Test Objective	Verify that the track chair can submit feedback to a reviewer for their review.
Procedures	<ul style="list-style-type: none"> <li>● Log in to the system with track chair credentials.</li> <li>● Navigate to the review section for a specific paper.</li> <li>● Click on the "ADD FEEDBACK" button for a reviewer.</li> <li>● In the "Write Feedback To Reviewer (optional)" field, enter feedback text (e.g., "Great analysis, but consider expanding on the methodology section.").</li> <li>● Click the "ADD FEEDBACK" button to submit.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>● The feedback is successfully submitted and saved.</li> <li>● The feedback is visible in the reviewer's profile or review section.</li> </ul>
Test Priority	Major
Date Tested	
Results	N/A*

\*: As of the time of testing, adding feedback to reviews is not implemented yet. However, it will be tested, before the demos and will be shown as a part of an example scenario.

Test ID	FT-TRACK_CHAIR-008
Test Type	Functional Test
Test Objective	Verify that the track chair can rate a reviewer's review on a scale of 0 to 10.
Procedures	<ol style="list-style-type: none"> <li>1. Log in to the system with track chair credentials.</li> <li>2. Navigate to the review section for a specific paper.</li> <li>3. Click on the "ADD FEEDBACK" button for a reviewer.</li> <li>4. In the "Give Rating To The Review" section, select a rating (e.g., "8") by clicking on the corresponding number.</li> <li>5. Click the "ADD FEEDBACK" button to submit the rating.</li> </ol>
Expected Results	<ol style="list-style-type: none"> <li>1. The selected rating (e.g., "8") is successfully submitted and saved.</li> <li>2. The rating is reflected in the reviewer's profile or review section.</li> </ol>
Test Priority	Critical
Date Tested	
Results	N/A*

\*: Please see the end of test with id “FT-TRACK\_CHAIR-007”

Test ID	UT-TRACK_CHAIR-009
Test Type	Usability Test
Test Objective	Verify that the track chair can view a reviewer's profile by clicking on the "Reviewer's Profile" link.
Procedures	<ul style="list-style-type: none"> <li>● Log in to the system with track chair credentials.</li> <li>● Navigate to the "CONFERENCE" section under "MY TASKS".</li> <li>● Locate the paper.</li> <li>● Click on the "Reviewer's Profile" link next to a reviewer's name (e.g., "Stephen Hawking")</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>● The reviewer's profile page opens in a new tab or within the same interface.</li> <li>● The profile displays relevant information about the reviewer (e.g., name, past reviews, ratings).</li> </ul>
Test Priority	Critical

Date Tested	27/04/2025
Results	Passed

Test ID	IT-TRACK_CHAIR-010
Test Type	Integration Test
Test Objective	Verify that the track chair can identify and manage pending reviews for the paper.
Procedures	<ul style="list-style-type: none"> <li>● Log in to the system with track chair credentials.</li> <li>● Navigate to the "CONFERENCE" section under "MY TASKS".</li> <li>● Locate the paper.</li> <li>● Check the status of the reviewer marked as "Pending".</li> <li>● Click on the "Assign More Reviewers" button to assign additional reviewers if needed.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>● The "Pending" status for reviewer is clearly displayed.</li> <li>● The track chair can assign additional reviewers if necessary, and the status updates accordingly.</li> </ul>
Test Priority	Critical
Date Tested	29/04/2025
Results	Passed

Test ID	FT-SUPERCHAIR-001
Test Type	Functional Test
Test Objective	Enable other users to access the conference by inviting them into that specific conference
Procedures	<ul style="list-style-type: none"> <li>● Log in to the system with superchair credentials.</li> <li>● Navigate to the "CONFERENCE" section under "MY TASKS".</li> <li>● Click the "Invite People" button on the screen.</li> <li>● On the pop-up menu, search for the people to invite, select them by clicking the checkbox.</li> <li>● Click the "Invite" button. Then see them on the people list.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>● Newly added people also should be visible on the screen of the superchair.</li> <li>● Users should access the page.</li> </ul>

Test Priority	Critical
Date Tested	28/04/2025
Results	Passed

Test ID	FT-SUPERCHAIR-002
Test Type	Functional Test
Test Objective	Assign superchairs to the conference
Procedures	<ul style="list-style-type: none"> <li>● Log in to the system with superchair credentials.</li> <li>● Navigate to the "CONFERENCE" section under "MY TASKS".</li> <li>● Click the “Assign Superchair(s)” button on the screen.</li> <li>● On the pop-up menu, search for the people to assign as superchair, select them by clicking the checkbox.</li> <li>● Click the “Add Superchair” button. Then see them on the people list as superchairs.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>● Superchair should see the other superchairs.</li> <li>● Newly added superchairs can have access to the privilege of the system.</li> </ul>
Test Priority	Critical
Date Tested	27/04/2025
Results	Passed

Test ID	FT-SUPERCHAIR-003
Test Type	Functional Test
Test Objective	Add track chairs for the conference.
Procedures	<ul style="list-style-type: none"> <li>● Log in to the system with superchair credentials.</li> <li>● Navigate to the "CONFERENCE" section under "MY TASKS".</li> <li>● Click the “Add Track chair” button on the screen.</li> <li>● On the pop-up menu, search for the people to assign into that track, select them by clicking the checkbox.</li> <li>● Click the “Add Track chair” button. Then see them on the track list.</li> <li>● Later, you can edit and configure tracks.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>● People assigned to that new track should access the pages to</li> </ul>

	review within that track.
Test Priority	Critical
Date Tested	28/04/2025
Results	Passed

Test ID	FT-SUPERCHAIR-004
Test Type	Functional Test
Test Objective	To view an overview of the conference.
Procedures	<ul style="list-style-type: none"> <li>● Log in to the system with superchair credentials.</li> <li>● Navigate to the "CONFERENCE" section under "MY TASKS".</li> <li>● Click the “Conference Overview” button on the screen.</li> <li>● Here you can see the date interval, names of the superchairs, participants, details of the tracks, submissions created and submitted within that conference.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>● You can see the summary of the conference on that page.</li> <li>● If you want, you can configure everything about the conference on that page.</li> </ul>
Test Priority	Major
Date Tested	
Results	N/A

\*: As of the time of the testing for the final report submission, conference overview functionality is not implemented yet.

Test ID	FT-SUPERCHAIR-005
Test Type	Functional Test
Test Objective	To configure the conference, edit anything related to conference information, access information, submission information, paper assignment, reviewing information, and make sure that everything is set.
Procedures	<ul style="list-style-type: none"> <li>● Log in to the system with superchair credentials.</li> <li>● Navigate to the "CONFERENCE" section under "MY TASKS".</li> <li>● Click the “Conference Overview” button on the screen.</li> <li>● Click the “Configure conference”.</li> </ul>

	<ul style="list-style-type: none"> <li>● Make necessary changes and click on the “Apply configurations” button.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>● You can see the summary of the conference on that page.</li> <li>● If you want, you can configure everything about the conference on that page.</li> </ul>
Test Priority	Critical
Date Tested	29/04/2025
Results	Passed

Test ID	FT-SUPERCHAIR-006
Test Type	Functional Test
Test Objective	To assign the papers on related reviewers within the selected track.
Procedures	<ul style="list-style-type: none"> <li>● Log in to the system with superchair credentials.</li> <li>● Navigate to the "CONFERENCE" section under "MY TASKS".</li> <li>● Click the “Assign Papers” button on the screen.</li> <li>● Select paper name or author based on your preferences, and assign selected papers to selected reviewers.</li> <li>● Make necessary changes and click on the “Add reviewers” button.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>● Reviewers can see the papers assigned to them on their account.</li> <li>● After review, you can see the reviews by selecting the paper and see the details.</li> </ul>
Test Priority	Critical
Date Tested	29/04/2025
Results	Passed

Test ID	FT-SUPERCHAIR-007
Test Type	Functional Test
Test Objective	To manage people on the tracks. Add, remove, edit, and give privileges for some users.
Procedures	<ul style="list-style-type: none"> <li>● Log in to the system with superchair credentials.</li> <li>● Navigate to the "CONFERENCE" section under "MY</li> </ul>



	<p>TASKS".</p> <ul style="list-style-type: none"> <li>● Click the “Add People to Track” button on the screen.</li> <li>● Add people as your preferences, then click on the “Accept” button.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>● Added people can see the papers related to that track.</li> <li>● Later, superchair can remove them from the track if something is wrong with them.</li> </ul>
Test Priority	Major
Date Tested	28/04/2025
Results	Passed

Test ID	FT-NOTIFICATIONS-001
Test Type	Functional Test
Test Objective	To verify the functionality of the notification system for invites, new paper submissions, reviewer information.
Procedures	<ul style="list-style-type: none"> <li>● Log in to the system with any user credentials.</li> <li>● Navigate to the "TOP BAR" section and click on the “NOTIFICATIONS” button.</li> <li>● You can view your notifications, and make adjustments based on them.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>● You can view your notifications, come from other users, and conferences.</li> <li>● You can delete, accept, and reject the notification based on its type.</li> </ul>
Test Priority	Major
Date Tested	20/04/2025
Results	Passed

Test ID	FT-COMMUNICATIONS-001
Test Type	Functional Test
Test Objective	To verify the functionality of the chat system for communication between superchair and other users.
Procedures	<ul style="list-style-type: none"> <li>● Log in to the system with any user credentials.</li> </ul>

	<ul style="list-style-type: none"> <li>● Navigate to the "CHAT" section in the top bar.</li> <li>● Select an existing user or start a new chat.</li> <li>● Send a text message and verify if it is delivered instantly.</li> <li>● Check if the recipient receives the message in real-time without refreshing the page.</li> <li>● Test sending an attachment (e.g., PDF, image) to the recipient.</li> <li>● Log in with the recipient's account to verify the received message and attachment.</li> <li>● Test typing indicators by typing a message and observing if the recipient can see "User is typing..." status.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>● Messages should be sent and received instantly without page refresh.</li> <li>● Attachments should be successfully sent and accessible to the recipient.</li> <li>● Typing indicators should appear correctly.</li> <li>● Chat history should be intact after re-login.</li> </ul>
Test Priority	Critical
Date Tested	23/04/2025
Results	Passed

Test ID	FT-CREATE-001
Test Type	Functional Test
Test Objective	Verify that a user can successfully create a new conference
Procedures	<ul style="list-style-type: none"> <li>● Navigate to the "Create Conference" screen.</li> <li>● Fill in all required fields (Conference Name, Acronym, Website, City, Country, etc.)</li> <li>● Configure Access Information (Double-blind review, PC Member visibility, etc.)</li> <li>● Configure Submission Information</li> <li>● Configure Reviewing Information</li> <li>● Click the "Create Conference" button.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>● The system successfully creates a new conference.</li> <li>● The newly created conference appears under "Upcoming Conferences."</li> <li>● A Success message is displayed to user</li> </ul>
Test Priority	Critical
Date Tested	27/04/2025

Results	Passed
---------	--------

Test ID	FT-CREATE-002
Test Type	Functional Test
Test Objective	Verify that a user can import settings and members from an existing conference.
Procedures	<ul style="list-style-type: none"> <li>● Click "Import from Other Conference."</li> <li>● Select an existing conference (e.g., BIL 2023).</li> <li>● Choose roles to import (Superchair, Track Chair, PC Members).</li> <li>● Click the "Import" button.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>● Selected roles are successfully imported into the new conference.</li> <li>● A confirmation message is displayed.</li> <li>● The imported roles appear in the respective sections.</li> </ul>
Test Priority	Major
Date Tested	
Results	N/A*

\*: As of the time of the testing for the final report submission, import functionality is not implemented yet.

Test ID	FT-CONFERENCE-004
Test Type	Functional Test
Test Objective	Verify that a user can edit conference details after creation.
Procedures	<ul style="list-style-type: none"> <li>● Open an existing conference.</li> <li>● Edit details such as Conference Name, Submission Web Page, Contact Emails, etc.</li> <li>● Save changes.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>● The updated details are saved successfully.</li> <li>● The modified information appears correctly in the conference details.</li> </ul>
Test Priority	Major
Date Tested	29/04/2025

Results	Passed
---------	--------

Test ID	IT-CONFERENCE-005
Test Type	Integration Test
Test Objective	Verify that users with different roles (Superchair, Track Chair, PC Member) can access only their permitted functionalities.
Procedures	<ul style="list-style-type: none"> <li>● Log in with different roles.</li> <li>● Attempt to perform role-specific actions (e.g., Track Chair assigns papers, PC Members review submissions).</li> <li>● Verify that unauthorized users cannot access restricted functions in conference.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>● Users can access only their permitted functionalities.</li> <li>● Unauthorized access attempts are denied with an appropriate message.</li> </ul>
Test Priority	Critical
Date Tested	01/05/2025
Results	Passed

Test ID	UT-CONFERENCE-006
Test Type	Usability Test
Test Objective	Verify the ease of navigation and accessibility in the conference creation process.
Procedures	<ul style="list-style-type: none"> <li>● Navigate through the "Create Conference" interface.</li> <li>● Check for clear labels, instructions, and tooltips.</li> <li>● Attempt to complete the process without prior guidance.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>● Users can navigate easily without confusion.</li> <li>● Labels and instructions are clear and informative.</li> </ul>
Test Priority	Minor
Date Tested	27/04/2025
Results	Passed

Test ID	FT-REVIEWER_CONFLICT_DETECTION-001
Test Type	Functional Test
Test Objective	Verify that the system automatically detects conflicts of interest when assigning reviewers to a paper.
Procedures	<ul style="list-style-type: none"> <li>● Log in as a Superchair or Track Chair.</li> <li>● Navigate to the "Assign Reviewers" section.</li> <li>● Select a paper for assignment.</li> <li>● Choose a reviewer who is also an author on the same paper or has a known conflict (e.g., same institution, past co-authorship).</li> <li>● Click "Assign Reviewer" and check if the system displays a conflict warning message.</li> <li>● Attempt to bypass the conflict and manually assign the reviewer.</li> <li>● Verify that the system prevents the assignment or requires explicit justification.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>● The system should automatically flag conflicts of interest when detected.</li> <li>● The user should not be able to assign a conflicted reviewer without providing a justification.</li> <li>● A warning message should appear explaining the reason for the conflict.</li> </ul>
Test Priority	Critical
Date Tested	
Results	N/A

\*: As of the time of the testing for the final report submission, conflict of interest functionality is not implemented yet.

Test ID	FT-USER_REGISTRATION-001
Test Type	Functional Test
Test Objective	Verify that users can successfully register on the platform.
Procedures	<ul style="list-style-type: none"> <li>● Navigate to the registration page.</li> <li>● Fill in name, email, password, and organization fields.</li> <li>● Submit the registration form.</li> <li>● Verify that a confirmation email is sent.</li> <li>● Click on the email verification link.</li> </ul>

	<ul style="list-style-type: none"> <li>Attempt to log in with the newly registered credentials.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>A confirmation email is sent.</li> <li>Clicking the email verification link activates the account.</li> <li>User should be able to log in successfully.</li> </ul>
Test Priority	Major
Date Tested	27/04/2025
Results	Passed

Test ID	FT-USER_ROLE_SWITCH-002
Test Type	Functional Test
Test Objective	Verify that users can switch between assigned roles (Superchair, Track Chair, Reviewer, Author) without losing session data.
Procedures	<ul style="list-style-type: none"> <li>Log in as a user with multiple roles (e.g., Superchair &amp; Reviewer).</li> <li>Enter a conference.</li> <li>Click on the role selection dropdown in the navigation bar.</li> <li>Select a different role from the dropdown (e.g., switch from Reviewer to Superchair).</li> <li>Verify that the dashboard updates with the correct role-based options.</li> <li>Switch back to the previous role and check that the correct data loads.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>Role change should reflect immediately on the UI.</li> <li>Users should not be logged out or lose session data.</li> </ul>
Test Priority	Major
Date Tested	
Results	N/A

\*: This functionality was initial

Test ID	FT-CONFERENCE-005
Test Type	Functional Test
Test Objective	Verify that Superchairs and Track Chairs can edit and delete conference tracks.
Procedures	<ul style="list-style-type: none"> <li>Log in as a Superchair or Track Chair.</li> </ul>

	<ul style="list-style-type: none"> <li>• Navigate to the Manage Tracks section.</li> <li>• Edit the track name and save changes.</li> <li>• Delete a track and confirm removal.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>• Tracks should be edited and deleted successfully.</li> <li>• Changes should reflect immediately in the conference settings.</li> </ul>
Test Priority	Major
Date Tested	28/04/2025
Results	Passed

Test ID	FT-CONFERENCE-006
Test Type	Functional Test
Test Objective	Verify that users can access the official website of a conference from the Conferencer platform.
Procedures	<ul style="list-style-type: none"> <li>• Log in as any user (Superchair, Track Chair, Reviewer, Author).</li> <li>• Navigate to the conference details page.</li> <li>• Locate the "Official Website" link.</li> <li>• Click the link and verify that: <ul style="list-style-type: none"> <li>○ A new tab opens.</li> <li>○ The correct conference website is loaded.</li> <li>○ There are no broken links or redirections to the wrong page.</li> </ul> </li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>• The conference website opens correctly in a new tab.</li> <li>• The website matches the expected URL provided by the Superchair.</li> <li>• No 404 errors, broken links, or incorrect redirections occur.</li> </ul>
Test Priority	Minor
Date Tested	29/04/2025
Results	Passed

Test ID	FT-CONFERENCE-007
Test Type	Functional Test
Test Objective	Verify that Superchairs can configure conference settings (submission deadlines, review policies).

Procedures	<ul style="list-style-type: none"> <li>● Log in as a Superchair.</li> <li>● Navigate to the Conference Settings page.</li> <li>● Change submission deadlines and review policies.</li> <li>● Save changes and verify that they persist after logout and re-login.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>● System should save and apply configuration changes.</li> <li>● Users should see the updated deadlines and policies.</li> </ul>
Test Priority	Critical
Date Tested	29/04/2025
Results	Passed

Test ID	FT-DOWNLOAD_PAPER-015
Test Type	Functional
Test Objective	Verify that reviewers and chairs can download submitted papers by clicking the paper icon.
Procedures	<ul style="list-style-type: none"> <li>● Log in as a Reviewer, Track Chair, or Superchair.</li> <li>● Navigate to the "My Tasks" or "Conference Submissions" page.</li> <li>● Locate a submitted paper in the list.</li> <li>● Click on the paper icon next to the submission.</li> <li>● Verify that: <ul style="list-style-type: none"> <li>○ The correct file starts downloading.</li> <li>○ The file format matches the original submission format (e.g., PDF).</li> <li>○ The file is not corrupted and can be opened successfully.</li> </ul> </li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>● Users with correct permissions can download the assigned papers by clicking the paper icon.</li> <li>● Unauthorized users cannot download restricted papers.</li> <li>● Downloaded papers are not corrupted and retain the original format.</li> </ul>
Test Priority	Critical
Date Tested	28/04/2025
Results	Passed

## 6. Maintenance Plan and Details



Maintaining the Conferencer application hosted on a personal Ubuntu VPS requires a combination of system administration, application monitoring, and security practices. Utilizing open-source tools, alongside established platforms like GitHub, can streamline these processes effectively.

## 6.1 System-Level Maintenance

- **Operating System Updates:** Regularly update the Ubuntu server's packages to patch security vulnerabilities and ensure system stability. This is typically done using `sudo apt update` & `sudo apt upgrade`. Schedule checks or perform manually at least weekly.
- **Server Access:** Secure Shell (SSH) is the standard for accessing the VPS command line. Ensure SSH access is secured, potentially by using key-based authentication instead of passwords, changing the default SSH port, and configuring `fail2ban` to block brute-force attempts.
- **Resource Monitoring:** Basic command-line tools like `htop` (for process and resource usage), `df` (disk free space), and `free` (memory usage) provide quick insights into server health. `netstat` or `ss` can be used to check network connections.

## 6.2 Application Stack Management (React, Flask, Nginx, MongoDB)

- **Process Management:**
  - **Flask Application:** Use a robust process manager like `systemd` to manage the Flask application service. Define a service unit file to handle starting, stopping, restarting, and ensuring the Flask app runs on boot. A WSGI server like `Gunicorn` or `uWSGI` should be used to run the Flask application efficiently behind the reverse proxy.
  - **React Frontend:** The React application should be built into static files (`npm run build` or `yarn build`).
  - **Reverse Proxy (Nginx):** As implemented, Nginx serves as the crucial reverse proxy, handling incoming requests, serving the static React build files, forwarding dynamic requests to the Gunicorn/uWSGI Flask backend, managing SSL termination (HTTPS), and potentially load balancing if needed in the future. Nginx configuration files (`/etc/nginx/sites-available/`) require careful management and testing (`nginx -t`).
  - **MongoDB:** Manage the MongoDB service also using `systemd`. Ensure it starts on boot and can be restarted reliably.
- **Containerization (Recommended):** As suggested by the architecture (Section 3.1.3), using `Docker` and `docker-compose` is highly recommended. This encapsulates each part of the stack (React build served by Nginx, Flask+Gunicorn, MongoDB) into separate containers defined in a `docker-compose.yml` file.
  - **Benefits:** Simplifies deployment, ensures consistent environments, makes updates easier (rebuild and restart containers), and isolates components.

- **Maintenance:** Involves managing the `Dockerfile` for each service, the `docker-compose.yml` file, and Docker volumes for persistent data (especially for MongoDB). Updates involve pulling new base images, rebuilding application images, and restarting the composition (`docker-compose down && docker-compose up -d`), often automated via CI/CD.

### 6.3 Database Maintenance (MongoDB)

- **Backups:** Regularly back up the MongoDB database. Use the `mongodump` utility to create binary backups. Schedule this using a `cron` job to run daily or weekly, storing backups locally (with rotation) and ideally also copying them to a remote, secure location (e.g., cloud storage). Test the restoration process periodically using `mongorestore`.
- **Monitoring:** Use MongoDB's built-in utilities like `mongostat` (real-time performance metrics) and `mongotop` (read/write activity per collection). Monitor disk space usage for the database files. Check MongoDB logs for errors or warnings (`/var/log/mongodb/mongod.log` by default).
- **Performance:** Regularly check index usage and consider adding new indexes based on slow query analysis (`explain()` command). Compact data files if necessary to reclaim disk space after large data deletions.

### 6.4 Monitoring, Logging, and Alerting

- **Application Logging:** Configure Flask's logging module to output detailed logs to files, including timestamps, log levels (INFO, WARNING, ERROR), and relevant context. Ensure logs are rotated (e.g., using Linux `logrotate`) to prevent them from consuming excessive disk space.
- **Centralized Logging:** For easier analysis, consider centralizing logs from Nginx, Flask, MongoDB, and the system itself. Open-source stacks like the `ELK Stack` (Elasticsearch, Logstash, Kibana) or `Loki + Promtail + Grafana` can aggregate and visualize logs.
- **Metrics Monitoring:** Tools like `Prometheus` (time-series database) combined with `Grafana` (visualization dashboard) provide powerful monitoring. Exporters can collect metrics from the Ubuntu host (`node_exporter`), Nginx, potentially MongoDB, and custom metrics from the Flask application.
- **Alerting:** Configure alerting rules based on metrics (e.g., high CPU/memory usage, low disk space, high error rates) or log patterns using tools integrated with your monitoring stack (e.g., Alertmanager for Prometheus, Grafana alerting).

### 6.5 Security

- **Firewall:** Configure the Uncomplicated Firewall ([ufw](#)) on Ubuntu to allow traffic only on necessary ports (e.g., 80 for HTTP, 443 for HTTPS, the SSH port). Deny all other incoming connections by default.
- **HTTPS:** Secure web traffic using HTTPS. Obtain free SSL/TLS certificates from Let's Encrypt using the [Certbot](#) tool. Configure Nginx to use these certificates and automatically redirect HTTP traffic to HTTPS. Ensure Certbot is set up to auto-renew certificates (usually handled by a systemd timer or cron job installed by Certbot).
- **Dependencies:** Regularly update application dependencies (Python packages via [pip](#), Node.js packages via [npm](#) or [yarn](#)) to patch known vulnerabilities. Utilize GitHub's **Dependabot** to automatically detect vulnerable dependencies and open pull requests to update them. Supplement this with manual checks using tools like [pip-audit](#) or [npm audit](#).

## 6.6 Version Control and Deployment

- **Version Control:** All application code (React, Flask) and infrastructure configuration (Nginx configs, Dockerfiles, docker-compose.yml, systemd units) are managed using [Git](#) hosted on **GitHub**.
- **Automated CI/CD:** The deployment process is fully automated using **GitHub Actions**. Workflows are configured to trigger on code commits to specific branches (in our case, [main](#)). These workflows perform the following steps:
  - Run automated tests (unit, integration).
  - Build the React frontend static assets.
  - Build Docker images for the Flask backend and potentially the Nginx frontend server.
  - Push the built Docker images to a container registry (e.g., Docker Hub, GitHub Container Registry).
  - Connect to the Ubuntu VPS via SSH.
  - Pull the latest Docker images.
  - Stop the running application containers ([docker-compose down](#)).
  - Start the application using the updated images ([docker-compose up -d](#)).
  - Perform any necessary database migrations.
  - Run post-deployment checks or smoke tests. This automated pipeline ensures consistent, reliable, and rapid deployments of new features and bug fixes.

This maintenance plan provides a foundation using common open-source tools, integrated with your existing GitHub workflow, suitable for managing the Conferencer application on an Ubuntu VPS. Regularity and testing (especially backups and security patches) are key to ensuring long-term stability and security.

## 7. Other Project Elements

### 7.1 Consideration of Various Factors in Engineering Design

During the design and development of Conferencer, several key factors were considered to ensure functionality, security, and sustainability while aligning with industry standards. Each factor's relevance to the system design is analyzed, along with its level of impact on a scale from 0 (none) to 10 (maximum effect).

#### **Public Health, Safety, Security, and Welfare**

##### **A. Public Health & Safety (Effect: 7/10)**

The system ensures data integrity and security, particularly for sensitive user information such as submitted papers, reviews, and conference decisions. Ensuring privacy and compliance with security standards like GDPR and KVKK prevents potential ethical and legal issues. Additionally, by offering cloud-based storage and redundancy mechanisms, data loss and corruption risks are minimized, ensuring that conference materials remain accessible to all users.

##### **B. Security (Effect: 8/10)**

Security is a critical aspect of the Conferencer system, considering that academic research submissions involve intellectual property and confidential peer reviews. Security measures include:

- Role-Based Access Control (RBAC) ensures that only authorized users (Superchairs, Track Chairs, Reviewers, Authors) access specific functionalities.
- Authentication via Google OAuth 2.0 and ORCID to prevent unauthorized access and account breaches.
- Session management features such as auto-logout after inactivity to prevent unauthorized access from shared devices.

##### **C. Welfare (Effect: 6/10)**

The system facilitates fair peer review by enforcing conflict-of-interest checks and transparent reviewer assignment policies. It ensures that reviews are based on merit and expertise, preventing biased evaluations.

Additionally, reviewers are provided with analytics to improve their review quality and track their contributions over time.

## **Global, Cultural, Social, Environmental, and Economic Factors**

### **A. Global & Cultural Factors (Effect: 6/10)**

Conferencer is designed for international academic conferences, accommodating a diverse range of users. The system supports:

- Flexible conference policies (single-blind, double-blind review options).
- User-friendly interface ensuring accessibility for users with different technical backgrounds.
- Potential future support for multi-language capabilities for better inclusivity.

### **B. Social Factors (Effect: 5/10)**

The system encourages collaboration among researchers by providing:

- Secure chat & messaging tools to facilitate discussions between chairs, reviewers, and authors.
- Transparent reviewer selection process ensuring fair and ethical paper evaluation.
- Metrics on reviewer performance, allowing conference organizers to reward and recognize reliable reviewers.

### **C. Environmental Factors (Effect: 4/10)**

The system reduces the need for physical documentation in academic conferences, minimizing paper waste and the carbon footprint associated with printing, mailing, and in-person meetings. Additionally:

- Hosting the system on cloud-based servers minimizes energy consumption compared to dedicated physical data centers.
- Optimized backend operations reduce redundant processing power usage, improving efficiency.

### **D. Economic Factors (Effect: 7/10)**

The platform is designed to be cost-effective, minimizing infrastructure expenses:

- Free-to-use software dependencies (React, Flask, MongoDB) reduce licensing costs.
- Self-hosted on a Linux VPS owned by the team to minimize operational costs.

- Optional AWS deployment for scalability, with controlled budget usage.

Factor Category	Specific Factor	Impact (0–10)	Key Design Considerations
<b>Public Health, Safety, Security, and Welfare</b>	Public Health & Safety	7	Data integrity, GDPR/KVKK compliance, cloud storage with redundancy to minimize data loss.
	Security	8	Role-Based Access Control (RBAC), Google OAuth 2.0 & ORCID authentication, session auto-logout.
	Welfare	6	Conflict-of-interest checks, fair reviewer assignment, reviewer analytics to improve quality.
<b>Global, Cultural, Social, Environmental, and Economic Factors</b>	Global & Cultural Factors	6	Support for diverse conference policies, user-friendly UI, planned multi-language support.
	Social Factors	5	Collaboration tools (chat & messaging), fair reviewer selection, reviewer performance metrics.
	Environmental Factors	4	Reduced physical documentation, cloud hosting efficiency, optimized backend operations to lower energy usage.
	Economic Factors	7	Free software stack (React, Flask, MongoDB), cost-efficient VPS hosting, scalable AWS option with budget control.

### 7.1.1 Constraints

#### Implementation Constraints

- The system will operate as a web application.
- React with TypeScript will be used for front-end development.
- Python Flask will handle business logic and RESTful APIs for the backend.

- MongoDB will serve as the primary database for storing conference, user, and submission data.
- The application will be deployed on a Linux VPS, managed by one of the team members, reducing dependency on external cloud services.
- AWS Cloud Services may be used as an alternative deployment platform if scalability demands increase.
- Docker will be used for containerization to maintain consistent environments across development, testing, and production.
- Docker Compose will orchestrate multi-container services.
- Google OAuth 2.0 and ORCID will be implemented for authentication.

### **Economic Constraints**

- The software dependencies used in the system are open-source and free to use.
- A self-hosted Linux VPS will be used as the primary deployment environment to minimize costs.
- If needed, AWS services may be used, but costs will be kept minimal by optimizing storage and compute resources.

### **Ethical Constraints**

- Users' personal data will be securely collected, processed, and stored following established security practices.
- The purpose of data collection and system processes will be transparently communicated to users.
- Users will have the ability to view, edit, or delete their personal information within the system.
- The project ensures compliance with GDPR (General Data Protection Regulation) and KVKK (Turkish Personal Data Protection Law) [1].
- Submitted papers will be protected from unauthorized access to prevent plagiarism and ensure research integrity.
- Conflict-of-interest handling mechanisms will be enforced during the review assignment process, ensuring fairness.

## **7.1.2 Standards**

The Conferencer project adheres to industry standards for software development, data privacy, web application design, database management, cloud deployment, and ethics.

### **A. Software Development Standards**

- a. ISO/IEC 12207 – Software lifecycle processes [2].
- b. IEEE 1012 – Verification and validation of system components [3].

### **B. Data Privacy & Security Standards**

- a. GDPR & KVKK Compliance – Ensures user data protection [1].
- b. ISO/IEC 27001 – Information security management system for ensuring data confidentiality and integrity [4].

### **C. Web Application Standards**

- a. RESTful API Conventions – Used for system interoperability.
- b. WCAG 2.1 – Accessibility compliance for user interfaces.

### **D. Coding Standards**

- a. Python Best Practices (PEP 8 & PEP 20) – Maintainable and scalable development.
- b. JavaScript & React Best Practices – Ensures frontend maintainability.

### **E. Database Standards**

- a. MongoDB ACID Transactions (where applicable) – Ensures reliable database operations.
- b. NoSQL Best Practices – Optimized for document-based data storage.

### **F. Cloud Deployment & Infrastructure Standards**

- a. Linux VPS Deployment – Ensures cost-effective and self-managed infrastructure.
- b. AWS Well-Architected Framework (Optional) – Used if cloud-based scaling is required.
- c. Docker & Kubernetes – Provides containerized deployment for portability and resource optimization.

### **G. Ethical Standards**

- a. ACM & IEEE Codes of Ethics – Ensures fairness, integrity, and user trust in research submissions and reviews [5].



## **7.2 Ethics and Professional Responsibilities**

### **7.2.1 Ethical Requirements**

The system must prioritize data privacy and security by ensuring that all user data, including personal and conference-related information, is collected, processed, and stored securely in compliance with regulations such as GDPR and KVKK. Following the ACM Code of Ethics, the platform ensures transparency and fairness in data handling and decision-making processes [4]. Submitted papers must be safeguarded against unauthorized access to prevent plagiarism and misuse. Transparency is essential, and the system must clearly communicate the purpose of data collection, storage, and processing to users while providing them with options to review, update, or delete their personal information. To ensure fairness and integrity, the platform should prevent conflicts of interest in the review process through automated conflict detection and anonymized reviewer and author identities where necessary. Accountability is also a critical aspect, and the system should maintain an audit trail of all actions, such as review assignments and decision-making processes, to ensure traceability. Finally, accessibility should be a guiding principle in the design, making the system inclusive for a diverse range of users, regardless of their technical expertise.

### **7.2.2 Professional Requirements**

In this project, professional requirements are closely tied to ensuring effective collaboration and maintaining clear communication among all team members. To achieve this, the team has adopted Jira for task management and tracking progress. Jira facilitates an organized workflow by allowing the team to create, assign, and monitor tasks systematically. It also helps ensure accountability, enabling members to focus on their respective responsibilities while maintaining transparency about the project's overall status.

Additionally, professionalism demands adherence to established coding standards and best practices. This includes writing clear, maintainable code, conducting peer reviews, and documenting the system effectively to ensure long-term sustainability and ease of future enhancements. Ethical use of data and compliance with applicable regulations, such as GDPR, further underline the professional obligations that guide this project.

## **7.3 Teamwork Details**

### **7.3.1 Contributing and functioning effectively on the team**

The project team functioned as a highly collaborative and adaptive unit throughout all phases of the development of *Conferencer*. Each member brought specialized skills to different parts of the system, while also contributing cross-functionally when required.

At the outset, the team collectively established the primary goals and milestones for the project, aligned with both the initial requirements and the evolving needs identified during development and testing. Task planning and tracking were systematically managed using Jira, which allowed detailed assignment of work items and facilitated agile planning. Additionally, the team held weekly in-person or remote meetings, where each member reported progress, identified challenges, and proposed solutions. This ensured alignment and accountability across all tasks.

Individual contributions included:

- **Abdurrahman Bilal Kar:** Led the system design, architectural design, deployment strategy, backend development, data model design, and CI/CD processes using Git. He also contributed to scalability planning and performance optimization.
- **Ahmet Memduh Tutuş:** Focused on the frontend, backend, and frontend-backend integration. He was responsible for database data arrangement and indexing strategies and contributed to testing, particularly functional and security tests.
- **Atilla Emre Söylemez:** Took charge of UI/UX design, worked extensively on the frontend, and played a key role in frontend-backend integration. He also participated in system testing and feedback loops.
- **Berkay Ayçiçek:** Led UI/UX design, contributed to the backend and data model development, and was responsible for the architectural convergence from Java Spring to Flask, ensuring smooth transition and integration of system components.

Task execution was balanced and adjusted dynamically based on evolving requirements, team members' workload, and challenges faced during the year-long development. When unanticipated changes arose, such as shifting from JWT-based authentication to a secure session-based authentication model to better suit our security and scalability needs, the team collaborated closely to update designs and redistribute tasks accordingly.

### 7.3.2 Helping creating a collaborative and inclusive environment

The team placed strong emphasis on creating a collaborative and inclusive working environment. This was achieved by:

- Ensuring equal opportunity for all members to propose solutions and voice concerns, particularly during architectural decision-making and testing feedback sessions.
- Encouraging knowledge sharing, especially when members worked outside of their primary responsibility areas. For example, frontend developers contributed to backend testing, and backend developers participated in UI testing.

- Utilizing pair programming and code review sessions on critical components such as the reviewer recommendation engine, session management, and the multi-track paper assignment system.
- Maintaining transparent progress tracking using Jira and meeting notes, so that all members had visibility into the project's status and challenges.

This collaborative approach not only facilitated smoother technical progress but also promoted personal and professional growth for all team members.

### **7.3.3 Taking lead role and sharing leadership on the team**

Leadership responsibilities were shared among team members according to the phase and technical demands of the project:

- During the architectural design and system modeling phases, Bilal assumed a leadership role, guiding the team through defining scalable, secure, and maintainable system architecture and data models. He was also responsible for distributing tasks among team members to ensure balanced workloads and effective progress.
- In the frontend and backend development phase, Memduh and Atilla took the lead in ensuring seamless integration and coordination between the client-side and server-side components.
- For the UI/UX design, Atilla and Berkay led the effort, translating user requirements and usability principles into a professional and accessible user interface.
- Berkay also led the design of the system's data models and contributed significantly to the backend architecture, ensuring consistency, scalability, and alignment with evolving requirements.
- In the testing phase, Memduh and Atilla coordinated testing activities and managed user feedback analysis and system performance evaluations.
- Bilal led the deployment strategy and execution, including Dockerization, ensuring the system was reliably deployed and maintained.

Each member also provided leadership within their respective areas of expertise, which fostered a decentralized and efficient decision-making process. The rotation of leadership roles contributed to a well-rounded development experience for the team and ensured no single point of dependency.

### 7.3.4 Meeting objectives

At the beginning of the project, the team defined both functional and non-functional objectives in the Requirements and Design phases. While many objectives were successfully implemented, some features were deferred or partially completed due to technical challenges and prioritization choices made during development.

Below is a breakdown of the key objectives and their final status, strictly aligned with the Final Report and Testing Results:

Objective	Final Status
User Authentication (Google, ORCID, Email/Password)	Fully implemented
Role-Based Access Control (RBAC)	Fully implemented
Session-Based Authentication (updated from JWT)	Fully implemented
Conference & Track Creation	Fully implemented
Program Committee Management	Fully implemented
Paper Submission & Tracking	Fully implemented
Reviewer Assignment System	Fully implemented
Conflict-of-Interest Detection	Partially implemented (marked as not implemented in testing results)
Reviewer Recommendation (Expertise/Keywords)	Fully implemented
Multi-Track Handling and Confidentiality	Fully implemented

<b>User Profile Management (Editable Fields, ORCID/Web of Science Links)</b>	Fully implemented
<b>Notification &amp; Communication System (In-app notifications, basic email alerts)</b>	Fully implemented
<b>Review Quality Ratings &amp; Reviewer Performance Metrics</b>	Fully implemented
<b>LLM-Generated Review Detection</b>	Deferred (design prepared but not implemented yet)
<b>Session Role Switching (Superchair/Reviewer/Track Chair)</b>	Fully implemented
<b>Paper Deletion &amp; Editing</b>	Fully implemented
<b>Reviewer Task Management (Sorting, Filtering)</b>	Fully implemented
<b>Download Assigned Papers</b>	Fully implemented
<b>Testing (Functional, Usability, Performance, Security, Compatibility)</b>	Most critical tests passed, except for features not implemented or deferred

#### **Non-Functional Goals:**

- **Usability:** Achieved — validated through usability testing and user feedback.
- **Performance:** Achieved — system met response time and stability benchmarks.
- **Reliability:** Achieved — confirmed by extended testing without critical failures.
- **Security:** Achieved — passed all critical security tests including injection, unauthorized access prevention, and secure session handling.

- **Scalability:** Achieved — the system successfully handled expected user and data load scenarios for small to medium-sized conference environments.

The team delivered the majority of functional requirements and met all core non-functional goals, even while navigating significant architectural changes such as moving from Java Spring Boot to Flask/MongoDB and replacing JWT authentication with a more secure session-based approach. Unimplemented or deferred features, such as the automated conflict-of-interest detection and the LLM-generated review detection system, were documented for future system iterations. The project’s success reflects not only technical proficiency but also effective teamwork and adaptive problem-solving throughout the development lifecycle.

### 7.4 New Knowledge Acquired and Applied

The development of *Conferencer* enabled the team to acquire and apply a wide range of new technical, design, testing, and project management skills. The team demonstrated adaptability by learning new technologies and strategies as the project evolved, especially when transitioning between architectural paradigms and integrating diverse system components.

#### Technical Knowledge

Area	New Knowledge Acquired	Learning Strategies
Full-Stack Development	Advanced React frontend (Hooks, Context API, dynamic state management); Flask backend (RESTful APIs); MongoDB data modeling and indexing	Online courses, official documentation, practical experimentation

<b>Security &amp; Access Control</b>	Transition from JWT to Session-Based Authentication, secure role-based access control (RBAC), and OAuth 2.0 (Google, ORCID) integration	Security tutorials, best practice guidelines, instructor feedback
<b>Database Management</b>	NoSQL schema design, data aggregation pipelines, indexing for performance optimization	MongoDB University resources, real-world trial-and-error, peer discussions
<b>DevOps &amp; Deployment</b>	Docker containerization, Nginx reverse proxy configuration, VPS hosting, CI/CD pipelines using Git	Docker and Nginx documentation, hands-on server deployment, team workshops
<b>Backend Architecture &amp; Scalability</b>	Modular Flask services, session management, API design principles, scalability testing	Flask official documentation, code reviews, scalability benchmark testing
<b>Reviewer-Paper Matching &amp; NLP</b>	Similarity analysis algorithms, keyword extraction, and reviewer-paper matching automation	Research articles, prototyping, algorithm testing
<b>UI/UX Design</b>	Responsive design principles, accessibility features, user-centric design patterns	Figma prototyping, user feedback, UX testing sessions
<b>Testing Methodologies</b>	Development and execution of functional, usability, performance, security, and compatibility test cases	Testing standards review, designing custom test plans, iterative testing

## Project Management & Collaboration

Area	New Knowledge Acquired	Learning Strategies
<b>Agile Project Management</b>	Sprint planning, Jira task management, Kanban workflow tracking	Using Jira for work item assignments, weekly team meetings for progress tracking

<b>Teamwork &amp; Leadership</b>	Task distribution, rotating leadership, workload balancing, and cross-functional collaboration	Weekly team syncs, peer mentoring during challenging phases, adapting leadership based on project needs
<b>Meetings &amp; Feedback Integration</b>	Refining requirements and architectural designs based on stakeholder feedback; evaluating design trade-offs	Iterative design reviews during instructor meetings; team decision-making sessions
<b>Task Tracking &amp; Adaptation</b>	Tracking weekly progress, identifying blockers early, and dynamically reallocating tasks based on team capacity	Jira updates, weekly retrospective discussions, problem-solving during integration

## Learning Strategies

Throughout the project, the team utilized a combination of structured and self-directed learning methods:

- **Official Documentation:** Extensively used for React, Flask, MongoDB, Docker, Nginx, and OAuth.
- **Online Courses:** To supplement knowledge in backend development, database optimization, and security.
- **Peer Mentoring:** Knowledge-sharing sessions within the team, especially during integration and testing phases.
- **Research and Prototyping:** Reviewing technical articles and experimenting with prototypes for complex features such as reviewer assignment systems.
- **Instructor Feedback:** Iterative design and testing refinements based on feedback from scheduled instructor meetings.
- **Hands-on Development:** Practical, iterative development and testing, learning by building and refining system components.



## 8. Conclusion and Future Work

### Conclusion

The *Conferencer* project successfully delivered a scalable, secure, and user-friendly conference management system that fulfills the majority of the functional and non-functional requirements defined at the beginning of the project. The system provides seamless workflows for user authentication, conference and track creation, program committee management, paper submission and tracking, reviewer assignment, and user notifications.

Through collaborative effort and adaptive development practices, the team navigated several technical challenges, including a major architectural transition from Spring Boot to Flask, a database migration to MongoDB, and a security model revision from JWT to session-based authentication. Core features such as reviewer assignment, multi-track confidentiality, user profile management, and role-based access control were fully implemented and validated through rigorous testing.

The team's focus on usability, scalability, and maintainability resulted in a system that meets the needs of small to medium-sized academic conferences. Feedback from testing, instructor meetings, and peer review was incorporated into the design and development cycles, contributing to the system's robustness and user-centric design.

While some features, such as automated conflict-of-interest detection and LLM-generated review identification, were deferred due to time constraints, the system architecture supports their future implementation without major redesign.

Overall, the project has provided valuable learning opportunities in full-stack development, database management, security, agile project management, and team collaboration, preparing all team members for future roles in the software engineering field.

### Future Work

Several enhancements have been identified for future iterations of *Conferencer*:

- **Conflict-of-Interest Automation:** Completing the development of automatic conflict-of-interest detection based on affiliations, co-authorships, and declared

conflicts.

- **LLM-Generated Review Detection:** Implementing the pre-designed framework for detecting AI-generated reviews using natural language processing techniques.
- **Advanced Reviewer Assignment:** Expanding the recommendation engine to incorporate more sophisticated reviewer-paper matching criteria and possibly integrating machine learning models.
- **Extended Notification System:** Improving the email notification infrastructure and adding real-time alerts across all user roles.
- **Scalability Enhancements:** Preparing the system for larger conferences with thousands of participants and papers by further optimizing database queries and backend processing.

These improvements will not only extend the system's capabilities but also enhance its usability, flexibility, and value to conference organizers and participants.

## 9. Glossary

**1 - GDPR:** The **General Data Protection Regulation (GDPR)** is a data protection law enacted by the European Union to protect individuals' personal data and privacy. It sets rules for how organizations collect, use, and store personal information of people in the EU.

**2 - KVKK:** The **Kişisel Verilerin Korunması Kanunu (KVKK)** is the Turkish law for the protection of personal data, similar to the GDPR. It regulates how personal data should be processed, stored, and protected by organizations within Turkey. The law aims to ensure privacy and give individuals control over their personal data.

## 10. References

- [1] TermsFeed, "KVKK vs. GDPR: Understanding Turkey's Data Protection Law," TermsFeed, 2024. [Online]. Available: <https://www.termsfeed.com/blog/turkey-kvkk-gdpr/>. [Accessed: Mar. 10, 2025].
- [2] ISO, "ISO/IEC 12207:2017 - Systems and software engineering — Software life cycle processes," ISO, 2017. [Online]. Available: <https://www.iso.org/standard/63712.html>. [Accessed: Mar. 10, 2025].

- [3] IEEE, "IEEE Standard for System and Software Verification and Validation," IEEE Std 1012-2016, 2016. [Online]. Available: <https://standards.ieee.org/ieee/1012/5609/>. [Accessed: Mar. 10, 2025].
- [4] ISO, "ISO/IEC 27001:2013 - Information technology — Security techniques — Information security management systems — Requirements," ISO, 2013. [Online]. Available: <https://www.iso.org/standard/43447.html>. [Accessed: Mar. 10, 2025].
- [5] ACM, "ACM Code of Ethics and Professional Conduct," 2018. [Online]. Available: <https://www.acm.org/code-of-ethics>. [Accessed: Mar. 10, 2025].