Bilkent University

Department of Computer Engineering

Senior Design Project

T2409

Conferencer

Project Specification Document

17.11.2024

Abdurrahman Bilal Kar

Ahmet Memduh Tutuş

Atilla Emre Söylemez

Berkay Ayçiçek

Supervisor: Can Alkan

Instructors: Mert Bıçakçı, Atakan Erdem

Innovation Expert: Muhammed Naci Dalkıran

Table of Content

1. Introduction

1.1. Description

Our senior project focuses on developing a more user-friendly conference management system designed to simplify the submission and review process for academic conferences. While offering features similar to EasyChair, HotCRP, and OpenConf, our system stands out with improvements in usability and more detailed statistics about users and reviewers. The system allows for easy management of conference series, multi-track events, and shared program committees while adding tools like reviewer recommendations, customizable profiles, and advanced filtering options. It's designed to meet the evolving needs of academic conferences, offering flexibility and detailed insights for organizers.

1.2. High-Level System Architecture & Components of Proposed Solution

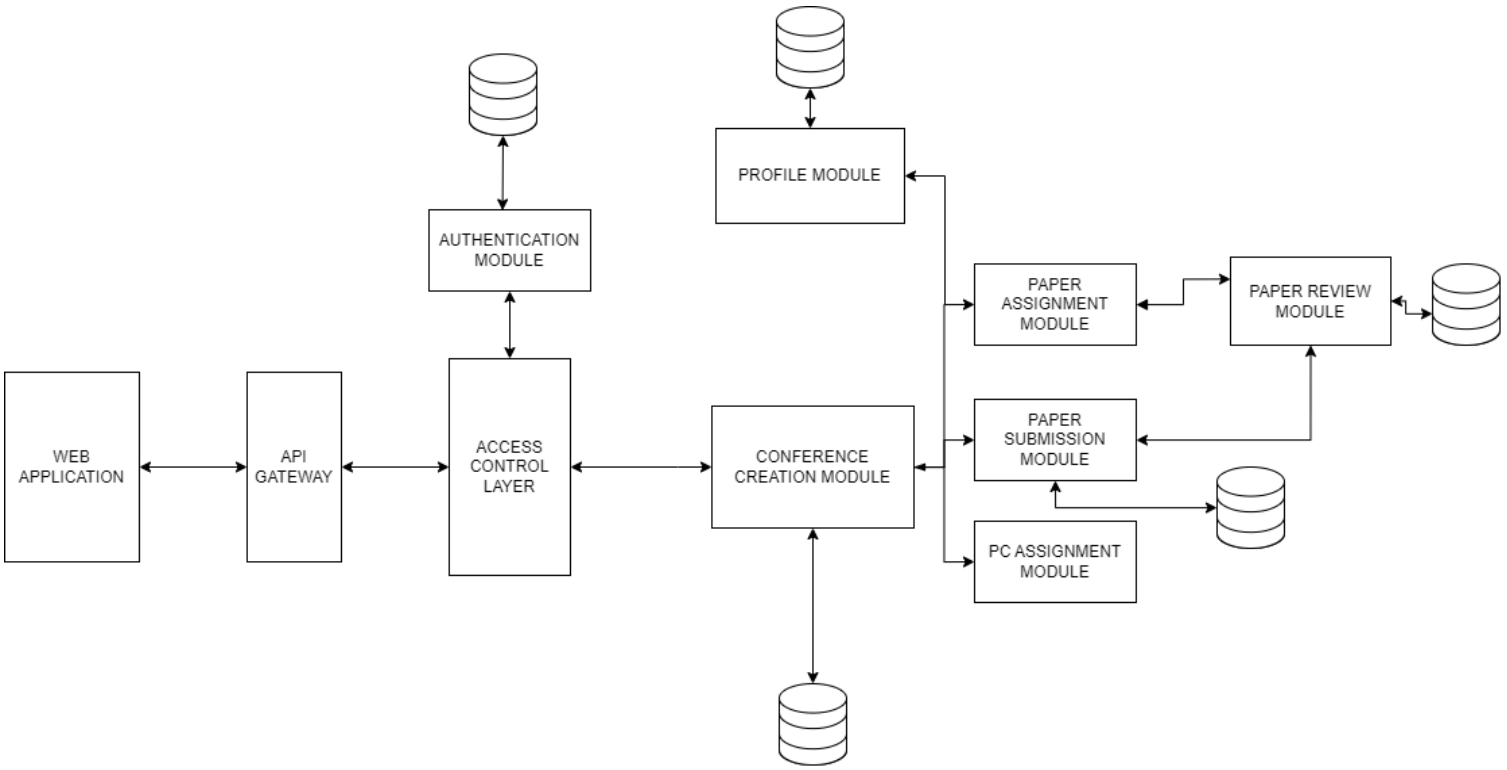

Figure 1. High-Level Architecture Diagram

1.3. Constraints

1.3.1. Implementation Constraints

- The system will operate as a web application.
- For the front-end development, **React** with **TypeScript** will be used.
- **Java Spring Boot** will be used to implement the business logic and RESTful APIs for the backend development.
- **Maven** will be used to manage dependency and automate project builds.
- **PostgreSQL** will serve as the primary relational database for data storage.
- The application will be deployed on **AWS** for cloud-based services.
- **Docker** will be used for containerization, enabling consistent environments across development, testing, and production.
- **Google OAuth 2.0** and **ORCID** will be used as login/register options.

1.3.2. Economic Constraints

- The dependencies on the system are free to use.
- If needed, payments may be made to AWS; however, the costs will be minimized.

1.3.3. Ethical Constraints

- Users' personal data will be collected, processed, and stored securely.
- The purpose of data collection and system processes will be explained clearly to the users.
- Users will have the ability to review. edit or delete their personal information.
- Privacy regulations such as GDPR and KVKK will be complied with.
- Submitted papers will be protected from unauthorized access to prevent plagiarism.
- The system will avoid conflicts of interest for the authors and reviewers in the review assignment process.

1.3.4. Sustainability Constraints

- Unnecessary performance overhead will be avoided to optimize server efficiency.
- A maintainable system will be developed to minimize future redevelopment costs.

1.4. Professional and Ethical Issues

1.4.1. Professional Issues

- Decisions about the project will be made by reaching an agreement with the group members.
- Source code will not be shared by any third party that is not relevant to the context of the development process.

1.4.2. Ethical Issues

- Users will be informed thoroughly about data collection and processing.

- No data irrelevant to the system will be requested from the users.
- The system will prevent cheating or conflicts of interest in the review process as best as possible.

2. Requirements

2.1. Functional Requirements

2.1.1. User Authentication and External Integrations

The system should support user authentication via native accounts, Google accounts, or ORCID accounts. This allows for easy login and secure profile management, with users having the ability to update their details as necessary. Additionally, reviewers should be able to link their profiles to their ORCID accounts for streamlined integration. The system must also support integration with Web of Science, enabling peer review recognition and external validation of review activities, enhancing the credibility and visibility of reviewers' contributions.

2.1.2. Program Committee Management

The system will make it easy for organizers to manage the Program Committee (PC). They can send invitations to potential members and quickly import PC members from previous years, saving setup time. Organizers will also be able to search and filter PC members based on keywords, review scores, and other criteria, helping them find the right people to review specific papers. This will help ensure that each paper is assigned to the best possible reviewers.

2.1.3. Conference Creation and Call for Papers

The system will allow organizers to set up new conferences by specifying details like the conference series, year, and track structure. Organizers can also generate a Call for Papers (CFP) that includes important information such as submission deadlines and requirements for each track. This feature simplifies the setup process and ensures that all necessary details are communicated clearly to potential participants.

2.1.4. Paper Submission & Tracking

The system streamlines the paper submission process, allowing authors to easily upload their manuscripts and accompanying materials through an intuitive interface. Submissions can be tracked, providing authors with updates on the status of their papers, from initial submission to final decision. The system will also support features such as automated compliance checks for formatting and file requirements, reducing administrative overhead. Reviewers and organizers can monitor submission progress, assign papers for review, and track key deadlines, ensuring a transparent and efficient workflow for all participants.

2.1.5. Multi-track Handling

The system supports single-track and multi-track conferences, offering flexibility to cater to diverse organizational needs. Each track can be managed independently in

multi-track conferences, or, as in the case of the Intelligent Systems for Molecular Biology (ISMB) Conference, a shared PC can be utilized across tracks. PC members in such setups can specify the maximum number of papers they agree to review, ensuring an equitable workload distribution. To prevent reviewer overload, track chairs will manage assignments within their respective tracks without direct access to specific papers assigned to PC members in other tracks. Instead, they will monitor overall review quotas, enabling a balanced and efficient review process while maintaining confidentiality across tracks.

2.1.6. Assigning Reviewer System

The system should include a recommendation engine that uses text analysis, such as keyword extraction and abstract similarity, to match papers with the most suitable reviewers. This feature will improve the efficiency of the review assignment process by leveraging expertise and reducing the manual workload for chairs.

2.1.7. Reviewing Assigned Papers

Conference chairs can view all reviews submitted within their conference (e.g., RECOMB 2025), while track chairs have visibility into reviews specific to their track (e.g., ISMB 2026 Sequence Analysis). Reviewers, or PC members, have personalized profiles where they can specify their reviewing preferences, including keywords to review or avoid and track-specific agreements for shared multi-track conferences. The system also automates conflict-of-interest detection by cross-referencing kinship statuses or past mentor-student relationships for registered users.

To enhance review quality, track and conference chairs have the option to rate reviews, helping identify reliable reviewers for future events. These ratings can optionally be shared across the same conference series but remain anonymized, ensuring that reviewers cannot access their own scores and that details are hidden from unauthorized users. Reviewers can also rate their peers' reviews on the same paper, further contributing to a culture of accountability and quality without compromising confidentiality.

2.1.8. User Profile Page

The system should provide a dedicated profile page for reviewers, allowing them to manage their information and preferences. The profile should include fields such as name, email, current and past affiliations, areas of expertise, keywords for topics they prefer to review or avoid, and conflict-of-interest declarations. Reviewers should also have the ability to specify track preferences for multi-track conferences. Some fields, such as name and expertise, may be optionally visible to conference organizers and chairs to facilitate reviewer selection, similar to platforms like ORCID or Web of Science. At the same time, sensitive details like conflicts of interest remain private. This functionality enhances transparency, ensures fair assignments, and streamlines the review process.

2.1.9. LLM-Generated Review Detection (Future Plan)

The system should include tools to identify and flag reviews generated by large language models (LLMs). Using natural language processing techniques, these tools will analyze review content for signs of automation or lack of originality, helping to ensure the quality and authenticity of submissions. This feature will support organizers in maintaining a fair and credible review process.

2.2. Non-functional Requirement

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.

3. Feasibility Discussions

3.1. Market & Competitive Analysis

Conference management systems like EasyChair, HotCRP, and OpenConf are widely used, but they have several limitations. EasyChair, for instance, has been criticized for its outdated user interface and increasingly complex workflows, which can make it difficult for new users to navigate efficiently. Additionally, its shift toward a more business-oriented model, including fees for larger events, has drawn concerns from the academic community [1]. Many systems struggle with usability, and outdated designs make them less accessible to new users. Updating interfaces and making workflows simpler can solve these problems [2]. These systems often lack advanced features such as AI-based reviewer recommendation tools, fraud detection for reviews, and detailed analytics for conference organizers. They also provide limited support for efficiently managing large, multi-track conferences. Conferencer aims to address these issues by offering a simpler, more intuitive interface, smarter tools for reviewer-paper matching, and advanced features designed for the needs of modern academic conferences.

3.2. Academic Analysis

This project builds on the academic principles of human-computer interaction and information retrieval to address the challenges in current conference management systems. Usability studies emphasize the importance of intuitive interfaces for improving user experience, especially in complex workflows like conference organization. Using AI in systems like Conferencer requires careful thought to ensure fairness and transparency. Following ethical principles will make these tools more trustworthy and effective [3]. Additionally, research in natural language processing and AI supports the use of automated tools for reviewer-paper matching and fraud detection, filling gaps in fairness and efficiency. By incorporating these advancements, this project contributes to the academic fields of system design and ethical AI, while providing practical solutions for modernizing conference workflows.

4. References

[1]Fournier-Viger, P. (2021). EasyChair: Is it still good? Retrieved from https://data-mining.philippe-fournier-viger.com/easychair-is-it-still-good.

[2]Okimoto, M. L., & de Oliveira, E. H. T. (2014). Evaluating Quality and Usability of the User Interface: A Practical Approach. In *Design, User Experience, and Usability. Theories, Methods, and Tools for Designing the User Experience* (pp. 311-320). Springer, Cham. Retrieved from https://link.springer.com/chapter/10.1007/978-3-319-07668-3_31.

[3]Brey, P., & Dainow, B. (2023). Ethics by design for artificial intelligence. *AI and Ethics, 4*(4), 1265–1277. Retrieved from https://link.springer.com/article/10.1007/s43681-023-00330-4.