

Student Conference Electronic Submission System

Tamás Rozsnyai*, Andrea Tamás*, Ervin Erőss†, Barna Séra† and Csaba Sulyok*

* Faculty of Mathematics and Computer Science, Babeş-Bolyai University
RO-400084 Cluj-Napoca, Romania

† Codespring

RO-400664 Cluj-Napoca, Romania

rozsthomas@gmail.com; tamasandii@gmail.com; eross.ervin@codespring.ro;
sera.barna@codespring.ro; sulyok.csaba@cs.ubbcluj.ro

Abstract—Scientific student conferences make it possible for undergraduate students to present their research in front of a committee of experts. These events usually use proprietary online solutions for submissions and paper management, making it difficult for students to move freely among them and always have up-to-date information.

The SConES project aims to ensure a single online platform for announcing and applying to student conferences. Although the authors use Transylvanian conferences as an inspiration point for functionalities, the proposed system may be used on an international scale. The target audience of the project consists of student conference organizers, applying students, their supervising professors/tutors, and committee members. The platform provides the opportunity to browse conferences and view detailed data. Students/authors are additionally able to view their submitted papers, follow status changes and edit their personal information.

The current paper outlines the architecture of the project, the communication between components, main functionalities with technologies/tools used during development.

Index Terms—student conferences; regional conferences; electronic submission system

I. INTRODUCTION

Scientific student conference activities have long played an important role in the lives of Hungarian university students [1]. The first such conferences are organized in the 1950-1951 academic year, inspiring a rising tendency towards further additions. They are often attributed the abbreviation TDK (*Tudományos Diákköri Konferencia*, or Scientific Student Conference). The first national event is held in 1955 with 109 presentations. In the following years, the event is divided into separate sections based on the primary addressed discipline. Since 2003, university students are able to submit student conference papers in 16 different sections of the National Scientific Student Conference (OTDK) [2].

The localized institutional student conferences provide an opportunity for aspiring bachelors of science to present their academic findings in front of a committee of experts. In addition to the professional feedback, a competitive edge is given to the events, as the best student papers have a chance to advance to the national phase. Attending such an event may be the first milestone in the academic pursuit of a student, providing valuable routine, possibly motivating an entry into higher academic circles, such as attempting an MSc or PhD program [3]. Another advantage may be the development

of effective communication and presentation skills; these are not only appreciated by higher academic officials, but also by students, as evidenced by a physics conference organized primarily by students in 1994 [4].

Besides numerous Hungarian cities, institutional TDKs are also organized over the borders, e.g. in Transylvania, Upper Hungary (presently a region of Slovakia), Transcarpathia and Vojvodina.

In 2010 a new online system is set up to ensure the smooth organization of the OTDK [5]. This system provides a centralized way of submitting papers, registration and keeping in touch. Most of the affiliated institutional TDKs also have separate portals; e.g. Transylvanian conference venues in Cluj-Napoca [6], Târgu Mureş [7] and Miercurea Ciuc [8], hosted by different universities all have their own online system. The separation of these both in style, layout and expected user input makes student interaction difficult when advancing from one conference system to another.

The current paper presents the Student Conference Electronic Submission System (SConES) project, whose main inspiration comes from the diversity of institutional TDK management. As the name of the project reflects, the goal is to create a unified interface through which applications for academic student conferences can be made. While the authors draw inspiration from the Hungarian conferences mentioned previously, general usability is kept in mind. The target audience is student conference organizers, applicants, authors, their supervising professors, mentors, tutors, and committee members.

The aim of the project is to make it easier for users to browse and apply for conferences. It also aims to provide a consistent aesthetic event browsing on any device; a responsive design provides mobile phone users a similar user experience to that of larger screens. An important aspect addressed in the design phase is maximal customizability; it is possible to specify the deadlines of the different tracks/sections within the events, to choose the scope and format of the abstract and the article individually. Opting for a streamlined experience, a “*convention over configuration*” approach is taken for these settings: the system uses default built-in configuration for any setting not provided. Students are allowed to submit papers as well as select co-authors and supervisors. It is possible to view the papers already submitted, and to download the

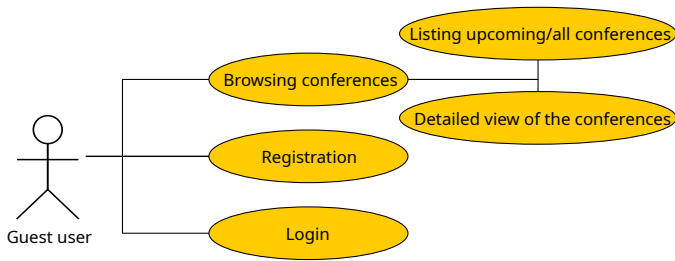


Fig. 1. Guest user functionalities

previously uploaded documents. Project functionalities also include automated e-mail sending when certain events occur, such as registration or successful submission of a paper.

Certain implementation ideas of the SconES project are inspired by the EasyChair [9] website. Operating since 2002, this platform is also used to advertise and apply for academic conferences, however it does not address the organizational specifics of student conferences: much like license theses/dissertations, the TDK research work and creative process is accompanied by the guidance and assistance of one or more supervising teachers and/or mentors.

The paper is structured as follows: Section II details the implemented functionalities, Section III describes the architecture of the project, covering communication and the data model. Section IV mentions the technologies and tools used during development, followed in Section V by a description of how the application works. Section VI provides conclusions and opportunities for further improvement.

II. FUNCTIONALITIES

The SConES system provides a unified platform for TDK organizers and participants, where applicant students can easily upload papers. Through the same portal, the conference paper may be approved by the supervisor(s) and automatically forwarded to any assigned committee members. Students can follow the status changes of their submission.

The current chapter covers the implemented functionalities for guests and registered users.

Guest user

As Figure 1 shows, a guest (not logged in) user can browse the conferences, separating between only the upcoming ones (i.e. those for which they can still submit a paper) and all public conferences. The guest user has access to the conference details on a dedicated customizable page. Here they can view the extended description, the period, location, the sections for which papers can be submitted, the submission deadlines, etc. The guest also can register or log in, with a successful registration prompting an e-mail notification.

Registered user

As shown in Figure 2, there are more functionalities for a logged-in user. They inherit general functionalities of a guest, but can also mark interesting conferences as “Followed”, allowing quicker compact visualization in a later session.



Fig. 2. Logged in user functionalities

They can also submit a paper for upcoming conferences, providing a mandatory section/track. The tracks may specify their own abstract length, the size and format of the paper, and even the submission deadline. If these restrictions are not given for a specific track, it inherits them from the conference. The submission itself requires a title, a short description/abstract, and attached the actual paper in the accepted file format. At least one user must be marked as a “Supervisor” and other users can be marked as a co-authors. Both types of involved users are notified after paper submission via e-mail. Any papers where the authenticated user is a co-author or supervisor shows up in a list on the profile page of the afferent conference, and these can be listed separately as well. Information about the submission and its current status is shown, taking on one of the fixed values “Pending”, “Approved”, “Accepted”, or “Rejected”. The paper itself can also be downloaded here.

The users can list and edit their personal data (e.g. name, university). Changing these becomes relevant when a student advances their year of study, changes their major or enrolls in a different university. When submitting a paper, the actual/current data about the user is stored therein.

III. ARCHITECTURE

The SConES software system consists of two main parts (see Figure 3): the client side boasts a web application usable through a browser, and in the back there is a central server responding to client requests, processing and storing data.

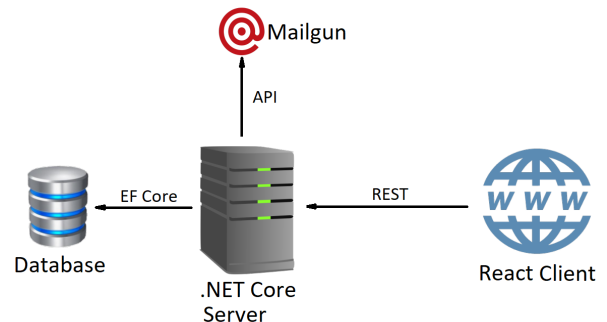


Fig. 3. Processes involved in the architecture, and the connections in-between

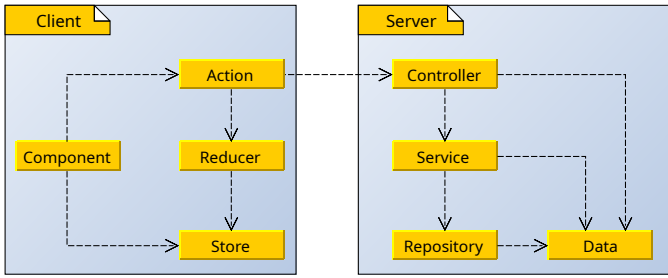


Fig. 4. Main components of both the client and server-side of the application

Communication between the two components is generally performed via HTTP requests; additionally the server sends e-mails to users through the external MailGun service.

Figure 4 shows the main components of the client and the server in detail, emphasizing the relationships with each other.

The client side takes an alternative approach to the typically followed MVC¹ patterns: the “Component” is related to both the “Store” and the “Action”—these are the interactive building blocks of the website. Actions communicate with the server and change the state of the Store through “Reducers”. Based on the data stored in the Store, components change their status.

On the server side, the “Repository” layer is responsible for database operations. It is called by the “Service” layer, which is responsible for business logic and any data manipulation. The “Controller” layer is located on the top level; it provides the entry points accessible to the client web application, acting as a proxy towards the services in the following tier. “Data” contains the various DTOs (Data Transfer Object) and domain model representations accessible throughout the backend.

Communication

Communication between the web client and the server takes place through RESTful services. REST (Representational State Transfer), described by Roy T. Fielding in his doctoral dissertation in 2000 [10], is an architectural model for implementing client-server communication. Services that meet the described principles are deemed RESTful. They make the data clearly available through a uniform interface, implement client-server based communication, and provide caching capabilities. They are also characterized by statelessness; the server does not store session information about any client. This statelessness contributes to scalability.

On the client side, each component has its own directory. Within the directories, the action folders contain the files with the `fetch` calls needed to upload the component data. With the results received from the server, an appropriate type of action is created, to which the appropriate reducer responds. The task of the reducers is to update the status of the application based on received information.

The ASP.NET Core used on the server side supports the creation of a web API that receives and serves requests from the client using controllers. Each model is attributed such a controller, with the implementation located

¹Model-View-Controller

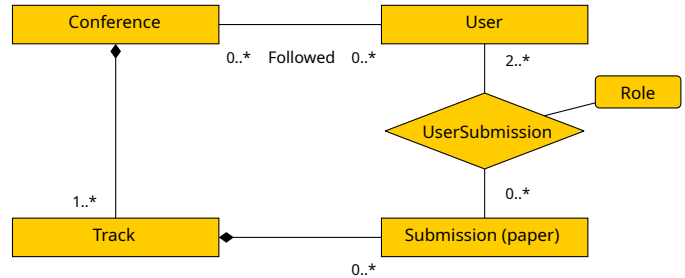


Fig. 5. Data model with entities and connecting relationships

in the `EDU.CS.TDK.Controllers` namespace. The class is marked with the `[ApiController]` and `[Route()]` attributes, mapping incoming requests to controller methods.

The data is passed as JavaScript Object Notation (JSON) objects: a human-readable key-value format for data exchange and storage. It is supported by all modern programming languages and is therefore practical for communication between two components written in different languages.

The requested information is sent to the client along with appropriate HTTP status codes.

E-mail sending mechanism

The scope of the project includes active informing and keeping in touch with the user, notifying them on certain events. For example, formatted personalized e-mails are sent when registering or submitting a paper successfully.

An e-mail sending system facilitates these functionalities, accomplished using the services of the MailGun server [11]. For the SConES project, connecting to the MailGun server is enabled by configuring the `SmtpClient` class of the `System.Net.Mail` namespace using the data in the `appsettings.json` file. The content of the messages is generated using internationalized templates. The language of the e-mail depends on the language selected by the user when registering on the online platform. After generating the appropriate e-mail body, the message is sent to the MailGun server, which then forwards it to the recipient.

Data model

The data model (see Figure 5) consists of four main tables and the relationship tables between them. The `Conference` table stores the conferences with afferent properties and information. The `Track` table stores associated sections and constraints where applicable. The `User` table contains user profile data and the role. The submitted papers are stored in the `Submission` table.

There is a one-to-many connection between the conferences and tracks. Each conference has at least one section, and each section belongs to only one conference due to individual constraints.

Multiple papers may be submitted for each section, while a paper can be submitted for a single section. In this case a one-to-many relationship between the track and submissions is used.

There is a many-to-many relationship between the submissions and users. A submission has several users involved, as each paper must have at least one author and at least one supervisor. And a user can submit (or supervise) multiple papers. These connections are implemented with an intermediate connection table (*UserSubmission*); the role of a given user in a given submission (“Author” or “Supervisor”) is also stored here.

There is also a many-to-many relationship between users and conferences, to represent the interest of the user in certain conferences by way of following.

Internationalization and localization

The SConES project aims to serve both Hungarian and international student conferences. This requires application globalization (g11n) [12]: a collective term for internationalization (i18n) and localization (l10n). The former allows the text of the online interface to be available in multiple languages; the latter is responsible for language conventions such as date and currency formats. After opening the online interface, the user can switch between the available languages, which are currently English and Hungarian.

The principle of internationalization is to store messages that vary by language in different resource files. As a result, switching between languages does not require recompilation. Thanks to this mechanism, the project can easily be extended with additional languages through the addition of JSON-formatted resource files.

IV. TECHNOLOGIES AND TOOLS

The server-side component of the project is realized in C# using the *ASP.NET Core* [13] cross-platform framework. To store the data, the *PostgreSQL* [14] relational database management system is chosen. The communication is realized through *Entity Framework Core*, which is an open-source, cross-platform framework implementing the object-relational mapping (ORM) paradigm.

The *React JavaScript* library is used to create the web client. It provides the ability to create independent and recyclable components that are typically written in an extended version of JavaScript, JSX, which incorporates an XML-like syntax for simple component description. *Redux* is used to easily manage the state of components. It is an open source JavaScript library whose main task is to manage the application state. It does this through a central store that is accessible to all components [15].

Globalization is enabled by the *react-18-next* package. The terms to be translated are specified as key-value pairs in JSON formatted resource files. The *Semantic UI* [16] framework is responsible for the aesthetics and responsive implementation of the application.

The project development workflow deploys the Scrum software development strategy. Git serves as a distributed version control system, for which GitLab provides the repository and project management interface. Continuous integration is

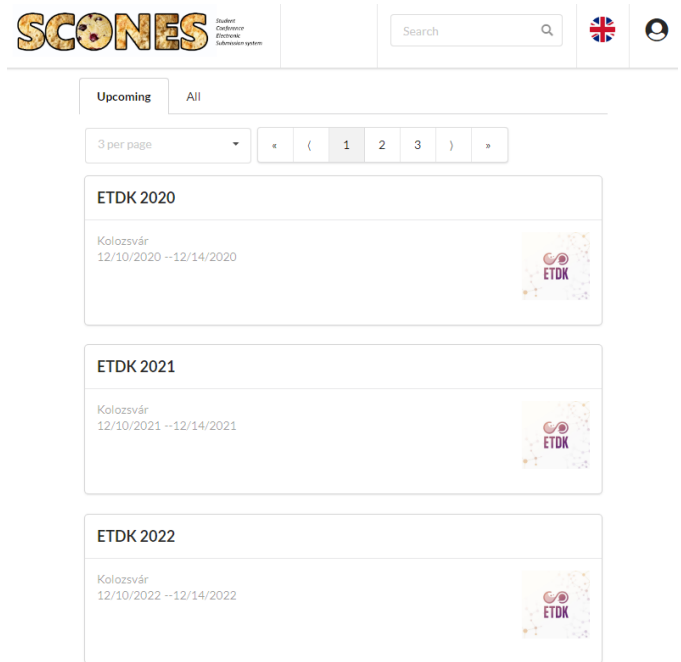


Fig. 6. The SConES main page: upcoming conferences with pagination

achieved using GitLab pipelines. Docker [17] and docker-compose are used for deployment and orchestration. To ensure code quality, static code analysis is performed using StyleCop on the server side, and ESLint on the client side.

V. USING THE APPLICATION

This chapter provides an in-depth explanation and illustration about the principal and more interesting use cases of the SConES system.


The main page of SConES consists of a list of “Upcoming” conferences (see Figure 6). To optimize network traffic and provide transparency, the number of conferences displayed at any one time is limited. The default setting is 3 conferences per page, modifiable by the user. Above the list is a tabular view allowing the user to change the listing between “Upcoming” and “All” conferences. Basic information is displayed for each conference: name, location, the time interval when it will be held, and a poster.

The header contains a search bar for filtering conferences by name, a drop-down menu for switching languages, and a profile icon, the functionality of which depends on whether the user is authenticated. From this drop-down menu, an authenticated user can access their papers, edit their personal data and log out. Guest users have the possibility to register or log in through this same menu.

Conference profiles are accessible by clicking any conference card. Besides basic information, these display the names of the sections with the submission deadline for each, and a detailed description of the conference. For logged-in users with previously uploaded papers, there is a table listing these with further details. In case the deadline has not yet expired,

SCONES 2021

Date of submission: 05/15/2020
Submission time: 18:38:34

Download your paper from here:  SConES.pdf
Conference: **ETDK 2021**
Track: computer science

Abstract

Scientific student conferences make it possible for undergraduate students to present their research in front of a committee of experts. These events usually use proprietary online solutions for submissions and paper management, making it difficult for students to move freely among them and always have up-to-date information. The SConES project aims to ensure a single online platform for announcing and applying to student conferences. Although the authors use Transylvanian conferences as an inspiration point for functionalities, the proposed system may be used on an international scale. The target audience of the project consists of student conference organizers, applying students, their supervising professors/tutors, and committee members. The platform provides the opportunity to browse conferences and view detailed data. Students/authors are additionally able to view their submitted papers, follow status changes and edit their personal information. The current paper outlines the architecture of the project, the communication between components, main functionalities with technologies/tools used during development.

Authors

Name	E-mail	Institution	Department	Specialization	Year of study
Andrea Tamás	andrea.tamas@gmail.com	Babeş-Bolyai University	Mathematics and Computer Science	Computer Science	3
Tamás Rozsnyai	rozsnyai.tamas@gmail.com	Babeş-Bolyai University	Mathematics and Computer Science	Computer Science	2

Supervisors

Name	E-mail	Institution	Department	Specialization
Csaba Sulyok	sulyok.csaba@gmail.com	Babeş-Bolyai University	Mathematics and Computer Science	Computer Science

Fig. 7. Details of the submission

the page contains a button which will navigate to the page for uploading papers.

The submission upload process consists of filling out a form in which tracked personal information about the user is automatically filled out. In addition, the user must select the appropriate section; this will impact the allowed length of the abstract and the format of the uploaded file. The user can specify co-authors, at least one supervising teacher, the title, the abstract and the submission file itself.

After submitting a paper, the user can access all its details including the current status. This takes on one of the following values: “Pending” (awaiting the approval of the supervisor), “Approved” (by the supervisor), “Accepted” (by the committee) or “Rejected”. The latter signifies the decision of the committee to decline the paper from the conference.

By clicking a certain submission, the user is navigated to the detailed page of the submission (see Figure 7). This page contains information about the other authors, the supervising teacher, and allows the re-download of the paper.

VI. CONCLUSIONS & FUTURE WORK

It is not easy for bachelor students engaged in scientific student activities to follow the different institutional pages involved in a multi-stage conference system. Differences, discrepancies or the lack of an online systems complicate the application process. The SConES project presented in this paper aims to unify these conferences through a shared platform for submissions and paper management. During development,

the authors have managed to create an online platform that can fill a gap in the management of Hungarian or even international TDKs. Using such a platform can make it easier for students to find and browse conferences, and as a result, organizers can more easily reach their target audience. Using an online system can also simplify the submission of papers and the communication.

Presently implemented and presented core functionalities include registration, submission and the management of submitted papers. The current section outlines some of the plans for further functionalities enhancing customizability and security.

Possibilities for further development include the search for conferences based on several different criteria, such as sections, date, name and location. The system would also allow organizers to create conferences, customize them, give students the opportunity to edit the abstract of a paper already submitted, upload a new document according to the rules of the given conference, or withdraw it if necessary. A further enhancement plan is the moving the supervising teacher approval process necessary in most TDKs to the online sphere. Also, committee members should receive the submissions through automated processes, allowing them to examine, grade, comment and provide feedback.

REFERENCES

- [1] S. Cziráki and P. Szendrő, “A TDK szerepe, szervezeti keretei és jövője a felsőoktatási tehetséggondozásban,” <https://bit.ly/2WVHhJh>.
- [2] Official webpage of the National Scientific Student Conference (OTDK). [Online]. Available: <http://otdk.hu/>
- [3] T. Larkin, “The student conference: A model of authentic assessment,” *International Journal of Engineering Pedagogy*, vol. 4, no. 2, pp. 36–46, April 2014.
- [4] M. Zadnik and A. Radloff, “A new approach to a communications unit: A student organized conference,” *A Focus on Learning*, pp. 292–296, 1995.
- [5] Online system of the National Scientific Student Conference (OTDK). [Online]. Available: <https://online.otdk.hu/>
- [6] Official webpage of the Transylvanian Scientific Student Conference organized in Cluj-Napoca. [Online]. Available: <https://etdk.kmdsz.ro/>
- [7] Official webpage of the Scientific Student Conference organized in Târgu-Mureş. [Online]. Available: <https://tdk.mmdsz.ro/>
- [8] Official webpage of the Scientific Student conference organized by the Sapientia Hungarian University of Transylvania in Miercurea-Ciuc. [Online]. Available: <https://tdk.cs.sapientia.ro/>
- [9] EasyChair webpage. [Online]. Available: <https://easychair.org>
- [10] R. T. Fielding and R. N. Taylor, *Architectural styles and the design of network-based software architectures*. University of California, Irvine Irvine, 2000.
- [11] Mailgun Official Page. [Online]. Available: <https://mailgun.com>
- [12] K. A. Mckethan and G. White, “Demystifying software globalization,” *Translation Journal*, vol. 9, no. 2, pp. 1–8, 2005.
- [13] A. Freeman, *Pro ASP.NET Core MVC*. Apress, 2016.
- [14] E. Geschwinde and H.-J. Schönig, *PostgreSQL developer's handbook*. Sams Publishing, 2002.
- [15] K. Chinnathambi, *Learning React: A Hands-on Guide to Building Web Applications Using React and Redux*. Addison-Wesley Professional, 2018.
- [16] Official Semantic UI documentation. [Online]. Available: <https://semantic-ui.com/>
- [17] D. Merkel, “Docker: Lightweight Linux Containers for Consistent Development and Deployment,” *Linux journal*, vol. 2014, no. 239, p. 2, 2014.