Extensible, cloud-based open environment for remote classes reuse*

Jakub Perżyło, Przemysław Watroba, Sebastian Kuśnierz, Jakub Myśliwiec, Sławomir Zieliński[0000−0002−0824−2608], and Marek Konieczny[0000−0002−1167−8568]

AGH University of Science and Technology, Kraków, Poland
{slawek,marekko}@agh.edu.pl

Abstract. The purpose of the presented work was to foster the reuse of whole remote classes, including the workflows involving multiple tools used during the class. Such a functionality is especially useful for designing new classes and introducing new topics into curricula, which happens quite often in laboratory-based courses in the Information Technologies area. The proposed approach allows professors to shorten the time it takes to create remote collaboration environments designed and prepared by their peers, by integrating multiple tools in topic-oriented educational collaboration environments. To achieve that, we split the system, code-named 'sozisel', into two parts - one of them responsible for facilitating creation of educational environments templates (models) describing the online class workflows composed of multiple tools, the other being responsible for executing the actual classes. For the environment to be extensible, we decided to use open, standards-based technologies only. This stands in contrast with most of commercially available environments, which are frequently based on proprietary technologies. The functional evaluation carried out on the sozisel prototype proved that the system does not require significant resources to be used in classes of medium size.

Keywords: cloud-based remote education, collaborative education, model driven architecture

1 Introduction

The proliferation of broadband Internet access, as well as the breakout of the COVID-19 pandemic, resulted in significant growth in usage of remote collaboration technologies in recent years. In face of the necessity to conduct remote classes, many teachers, including university professors, began to use environments capable of organizing remote meetings much more frequently than they

* The research presented in this paper has been partially supported by the funds of Polish Ministry of Science and Higher Education assigned to AGH University of Science and Technology.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_58

used to do before. Commercial companies such as Zoom, Cisco, Google, and Microsoft are major representatives of remote meeting environments vendors, and deliver mature products facilitating the remote meetings. The default set of tools used during the meetings can be enhanced with many plugins dedicated for the platforms. Due to business reasons, the vendors are rather reluctant to integrating their environments with external tools. Although it can be possible to call into the video conference from an external tool, the more sophisticated features are typically not exposed to the outside world. Free versions of the products are frequently limited in terms of functionality and number of session participants. Open source video conferencing solutions, such as Jitsi Meet or Jami, provide an interesting alternative at least from a researcher point of view by being open to modifications and extensions.

The work presented in this paper was motivated by the specific needs of IT education area. In AGH University practice it is quite common to update courses by developing new content for laboratory classes. Such an update is typically conducted by a small group of people, who lead the classes initially, and later share them with others. By sharing we mean not only passing on the static content (written instructions, etc.), but also the way of presenting the topics to a group of students, instructing about best practices, typical caveats, etc. Note also that subsequent occurrences of the newly developed labs may be distant in time, which impedes class sharing. We also observed similar needs among high school teachers, who cooperate with universities in the Małopolska Educational Cloud project (refer to [7] for an overview of the project and collaboration patterns).

Our idea was to build an environment that would facilitate two aspects of IT education: (1) sharing both the curricula, and sets of educational tools between professors, and (2) making the classes more repeatable by preparing, storing, and executing a workflow for a lab class. To make it possible we decoupled the processes of describing the class workflow and executing it. As a result of the definition phase, a model for the class is created, including the tools to be used during the class, the input data (files, links, etc.) for the activities to be performed during the class, and the tentative schedule of events (e.g. start of a survey). The model is converted into an executable workflow when a class execution phase starts.

The structure of the paper is as follows. Section 2 surveys the important developments in the subject area. Section 3 analyzes the requirements for sharing and reusing online sessions and describes the approach proposed in this article, which results in creation of reusable workflows. Section 4 discusses the results of evaluating a proof of concept implementation of the system code-named 'sozisel', which implements the discussed functionality. Section 5 concludes the paper and points out the directions of future work.

2 Related Work

Many universities and commercial companies provide trainings to online communities. There are many motivations behind that: to reduce costs, to improve

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_58

curricula, or to increase their reach [5,6,4]. In the latter case, there are two common options – either to support self-service learning (e.g., by following the Massive Open Online Courses (MOOCs) paradigm[1] [2]), or to follow a train-the-trainer approach.[3] In either case, the processes conducted in the learning environment need to be repeatable and predictable. Wide adoption of online teaching by educational institutions makes the requirement even stronger.

The predictability and repeatability requirements which apply to online environments consisting of various tools are especially important when teaching technical courses. Arguably, practical verification and experimentation in teaching engineering and computational sciences is crucial [1,3] for the students to succeed. Therefore, teaching technical courses online involves not only collaboration-related tools, but also virtualized lab environments[2]. As a common approach, before an environment is made available to a wide community, it is tested with regard to its functionality and appropriate resources needed by the environment are provided, so that the self-service users' experience is satisfactory.

In comparison with the self-service case, little attention is paid to systematically prepare teacher-led remote classes. Platforms focused on technology-oriented trainings provide extensive curricula, powerful tools[4] [5], and flexible virtualized lab environments[6] [7], but they lack of online session templates. Courses offered according to the train-the-trainer approach are supported with tools which commonly do not form an integrated environment. Reuse of the already completed class sessions also suffers from little support. In this paper we propose an approach which leverages open tools to provide an added value by allowing the trainers to share complete environments for leading classes instead of sharing class instructions only.

3 Sharing and reusing online sessions

In this section we describe our approach to supporting educational communities by providing means for structured composition of online class workflows. We assume that the communities are composed of teachers who provide similar, or at least related, courses to their respective students, and are willing to share their how-to knowledge within the community. We start from analyzing their requirements and follow with describing and justifying the main architectural elements of of the proposed online class reuse system.

[1] edX, `https://www.edx.org/`
[2] Coursera, `https://www.coursera.org/`
[3] Cisco Networking Academy Program, `www.netacad.com`
[4] Cisco Packet Tracer, `https://www.netacad.com/courses/packet-tracer`
[5] Mininet, `http://mininet.org`
[6] Cisco DevNet `https://developer.cisco.com`
[7] P4 behavioral model version 2, `https://github.com/p4lang/behavioral-model`

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_58

3.1 Actors and requirements

When a new content in the form of laboratory classes is introduced into a course, typically a single person or a small group of people are responsible for design and initial evaluation of the new content. After that, the content is shared with a larger group of teachers who conduct the classes. In case of classes led remotely, the sharing process can be supported by a structured feedback, e.g., by recordings of past sessions and written (or recorded) notes. The process is depicted in figure 1.

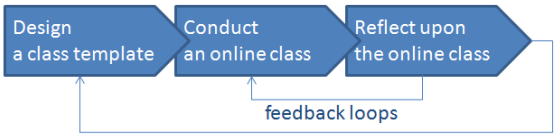

Fig. 1. Online class experience sharing idea.

Analyzing raw recordings is a time consuming task. Moreover, the person who attempts to lead a new class based on such materials only, needs to create a new environment practically from scratch, by instantiating and configuring the tools, etc. The presented system provides a remedy for the mentioned obstacles by focusing on sharing not only static materials, but the whole teaching expertise, including support for:

1. reproducing online sessions,
2. customizing the sessions to the needs of the teacher,
3. reflecting upon past sessions.

In the proposed approach, users acting as 'Designers', who produce the reference templates for classes, are aware of how to teach the respective subjects, what tools should be used and in what sequence, etc. They are also able to allot time for specific activities and include the assignments in the templates. After a template for a specific class is designed and shared, the users acting as 'Teachers' use it for instantiating actual classes. Typically, the people who design and develop the class, are teaching a few groups of students, and later share their experience with others. Therefore, the roles of 'Designer' and 'Teacher' are not disjoint. The template sharing and class instantiation processes are depicted in figure 2.

Note that in the presented process a Teacher is able to re-produce a class based either on a template, or on a previously stored instance. That directly reflects the observation of the behaviours of the teachers involved in the Małopolska Educational Cloud – they prefer to reuse the sets of tools and resources they used before over templates prepared by others. Simply put, they feel confident when using the services and content again.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_58

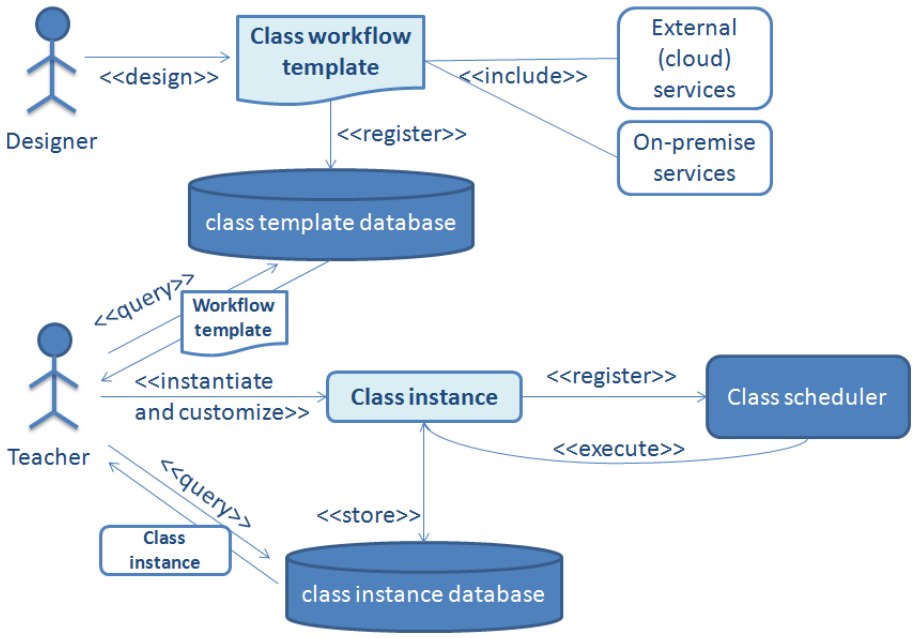

Fig. 2. Cooperation between users sharing class workflows.

3.2 Analysis of requirements

The requirements related to sharing expertise can be analyzed with regard to their statics and dynamics. In short, the former reflects the traditional way of sharing course materials, while the latter involves the use of environment-managed workflows.

Support for reproducing the sessions in what we call 'static aspect' involves re-creation of the sets of tools used during the sessions themselves, without requiring detailed knowledge of the tools' configuration, requirements, etc. That could be achieved by incorporating tools into an integrated environment that is able to instantiate them on demand. Customization options include the ability to provide different input (e.g., questions for a quiz) each time the online session is re-created. Reflection includes the ability to provide comments after completing the class. Both customization and reflection need to be addressed by introducing appropriate records (which refer to respective resources) in the the instance database. We call such records 'session resources'.

Regarding the dynamics of the class, the respective requirements can be elaborated as the following characteristics of remote class workflows. First, the difference between reproducing and simply replaying an online class should be pointed out. By reproducing we mean using the same tools (possibly in the same order), but not necessarily with the same time frames and using the same resources (e.g., displaying the same content). In this way the environment would

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_58

adapt to a particular person's style of teaching. Note that with regard to customization, the class workflow should be completely modifiable, it should allow introducing new tools, reordering of events, and even omitting some steps. Of course, after modification the workflow should not overwrite the original one, but be kept as a separate entry in the class instance database. Finally, with regard to reflection – it is safe to assume that in reality that activity is rarely performed just after the class is finished. Nonetheless, given the sequence of events that occurred during the class, the environment should be able to perform some automatic operations, including: (1) synchronizing the recording to be saved with the log of events contained in the class instance workflow, (2) automatic post-processing of service usage results in a service-dependent way (e.g. blurring faces, etc.). Such functionality will make a later reflection easier.

3.3 Architecure and implementation overview

The basic building blocks of the proposed environment is depicted in figure 3. The environment integrates cloud based and on premise services and exposes the session design, creation and management interfaces in the form of web applications. A database storing session templates and recordings serves as a backend.

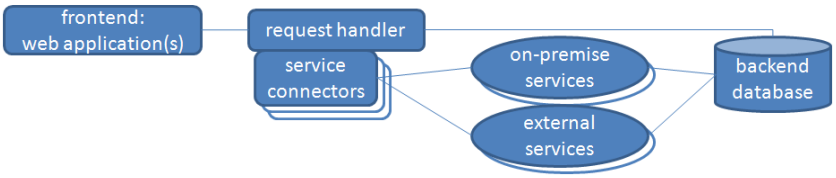

Fig. 3. Basic building blocks of the proposed environment.

Cloud services are referenced using software connectors that leverage their respective APIs. For consistency, we assume the same model of integration with local (on premise) services. Such a modular design provides for extensibility, which is needed to incorporate new, topic-specific services. Note that the usage of the backend storage is coordinated by the request handler, which is responsible for storing the events that are reported by multiple services in a consistent, synchronized way. That provides for easier review and annotation of past classes.

The prototype described in this paper integrates the following services: (1) audiovisual connectivity service, (2) quizzes and surveys service, and (3) shared whiteboard service. These correspond to the respective audiovisual service, virtual desktop service, and virtual whiteboard service described in [7], which describes the functionality of an environment built for MEC using commercial products. Regarding the most important implementation technologies for the prototype of the environment, we chose the following ones:

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_58

– Jitsi Meet[8] for audivisual connectivity between participants,
– Nginx[9] as a reverse proxy for frontend calls,
– PostgreSQL[10] for the (backend) instance database,
– Phoenix Framework[11] for server-side model-view-controller pattern implementation,
– GraphQL[12] as a basis for our API calls,
– React[13] for building frontend web applications.

Figure 4 illustrates how the chosen technologies fit into the environment architecture. The main criteria for the enumerated choices included their openness, popularity and versatility.

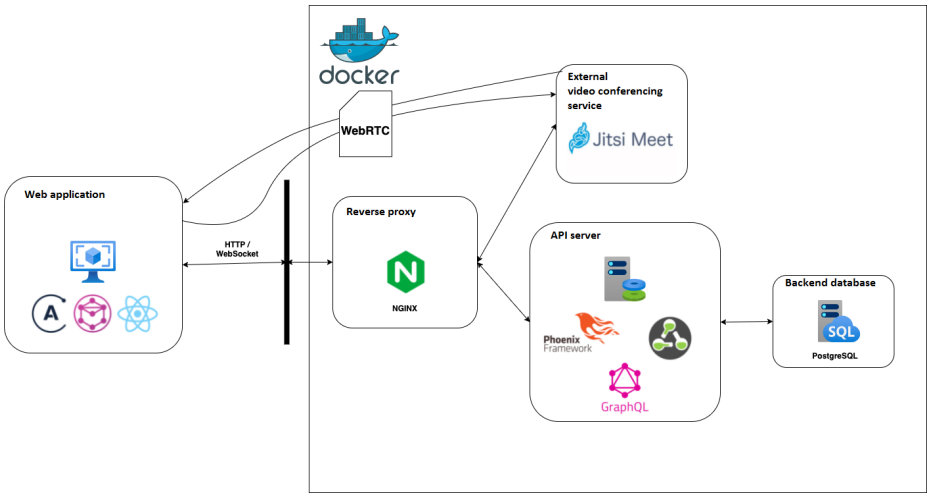

Fig. 4. Technologies used for prototype implementation.

The prototype implementation does not ideally fit the architecture. For the sake of implementation simplicity, we decided not to store the video session recordings inside the database, but rather hold references to their locations. The logical model of the database (simplified) is illustrated in 5.

The database is logically split into two parts, related to class templates and instances, respectively. A user who plays the Designer role, creates templates for online classes and provides the required session resources, references to which are held in the 'session templates' and 'session resources' tables, respectively. As

[8] Jitsi Meet, `https://jitsi.org/jitsi-meet`
[9] Nginx, `https://nginx.org`
[10] PostgreSQL, `https://www.postgresql.org`
[11] Phoenix Framework, `https://www.phoenixframework.org`
[12] GraphQL, `https://www.graphql.org`
[13] React, `https://reactjs.org`

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_58

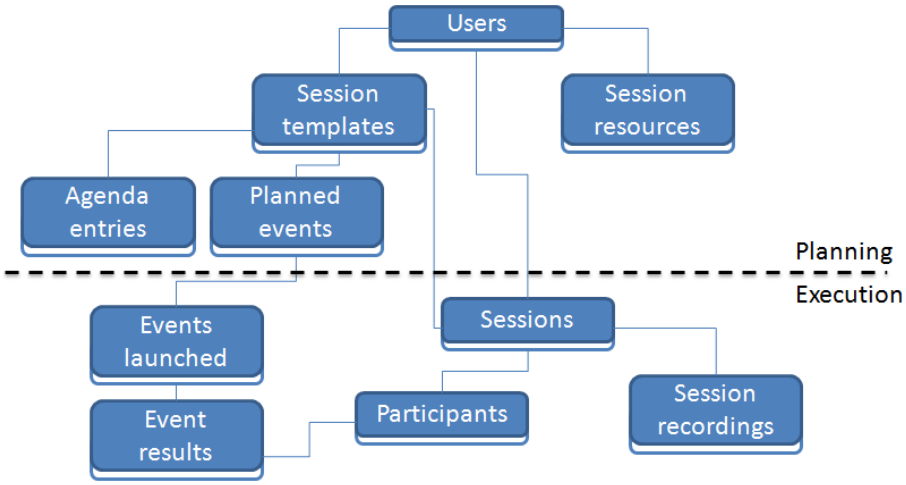

Fig. 5. Main tables of the backend database.

mentioned in section 3.2, they include, e.g., documents to be presented, questions for quizzes or surveys. Additionally, the Designer provides an agenda for the session in a textual form (table: 'agenda entries') and tentative timeline (table: 'planned events'). Based on that information, a user playing the Teacher role is able either to schedule a planned session directly, or to clone a session definition and customize it before scheduling (table: 'sessions').

When a scheduled class is conducted, the resources and tools are ready to be used by the session participants at the times defined by the schedule. However, it is up to the session leader (role: Teacher) if the timeline is followed strictly. The Teacher has an option either to delay, speed up or omit some planned events, depending on the students' learning paces or any other factors. The actual sequence of events is recorded in the 'events launched' table so that the users have an automatically recorded feedback, which can be used to modify the designs of future classes. Moreover, the video stream of the sessions are recorded (table: 'session recordings'), so the event log can be reviewed in synchronization with the recording, which can be useful from didactic point of view. Finally, the results of using the respective tools are kept in the 'event results' table.

4 Proof of concept evaluation

The sozisel prototype that was implemented used Jitsi Meet for video conferences. Additionally, it was capable of providing quizzes, surveys, and whiteboards. The services were hosted on premise. Figure 6 contains a screenshot of the project welcome page. The prototype is available for experiments at `https://sozisel.pl`. At the time of writing the article there was no English

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_58

version of the system. However, for the sake of readability, the screenshots include automatically translated text.

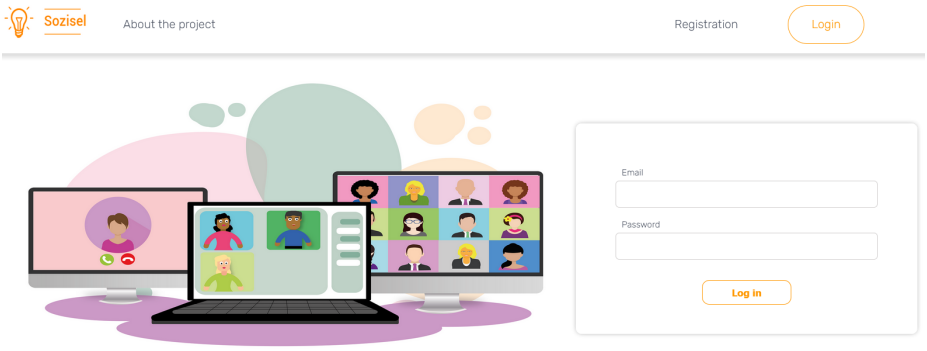

Fig. 6. Sozisel.pl welcome page.

During the tests we focused on both the design and instantiation phases of the online sessions. We started from template design (figure 7). In this use case, the Designer first creates a template, names it, and provides initial agenda. Then, tools to be used are selected (the screenshot contains forms that refer to a quiz, a questionnaire, and a whiteboard). Additionally, a tentative workflow is designed by setting tool usage start times as offsets from the start of the class, as well as respective durations. There is also an option to provide initial contents to be displayed by the tools.

The session scheduling form is depicted in figure 8. In the presented use case, the Teacher instantiates a session based on a template created and shared (made public) by a Designer. The teacher is able either to schedule the session in accordance to the designed workflow, or to copy the design and modify it before the session is executed.

Figure 9 shows a Teacher view of an ongoing session. In the screenshot, a quiz has just ended and therefore the grades are displayed by the Teacher's application. Because the quiz was the last planned activity, the application displays also an 'end session' button at the bottom of the screen.

The prototype was tested on a PC equipped with 6-core Intel Core i5-9600K CPU@3.70GHz with sessions for up to 25 users. During the sessions, all of the implemented tools were used. In the largest sessions the CPU load (average, calculated for a single core) was 3% for backend, 0% for frontend, and 200% for the video session (Jitsi). The users did not suffer from any lags resulting from the CPU load, and based on the numbers it is safe to assume that the same machine could host another session for a second 25-person class.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_58

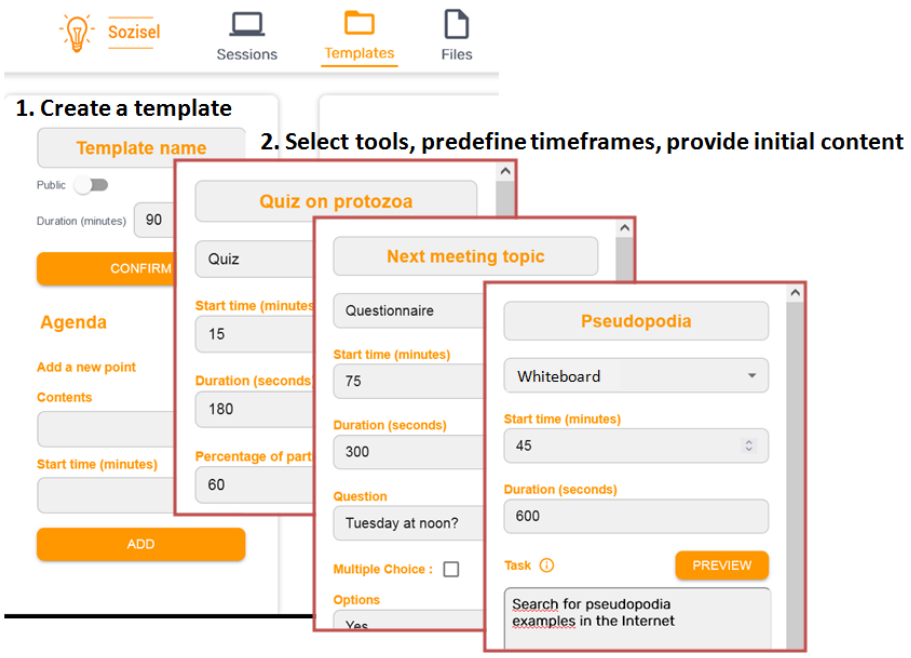

Fig. 7. Sozisel event planning phase.

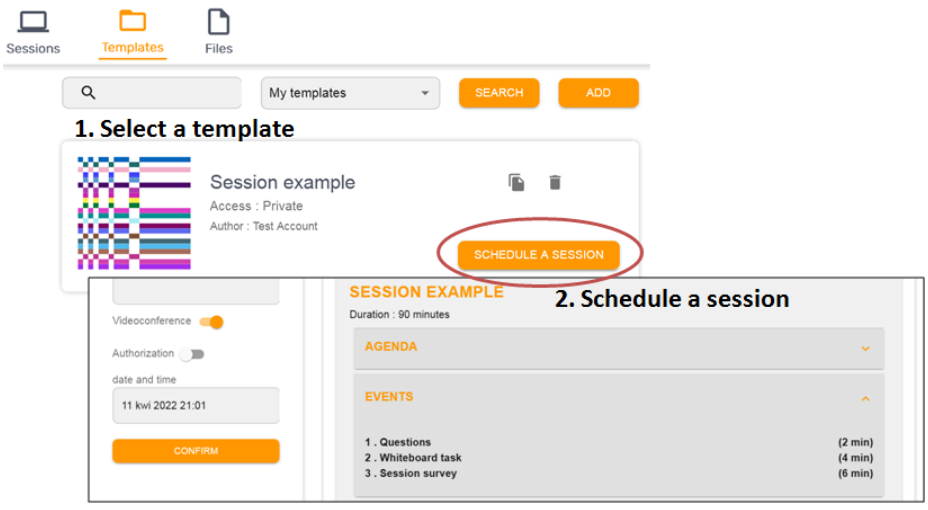

Fig. 8. Session scheduling

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_58

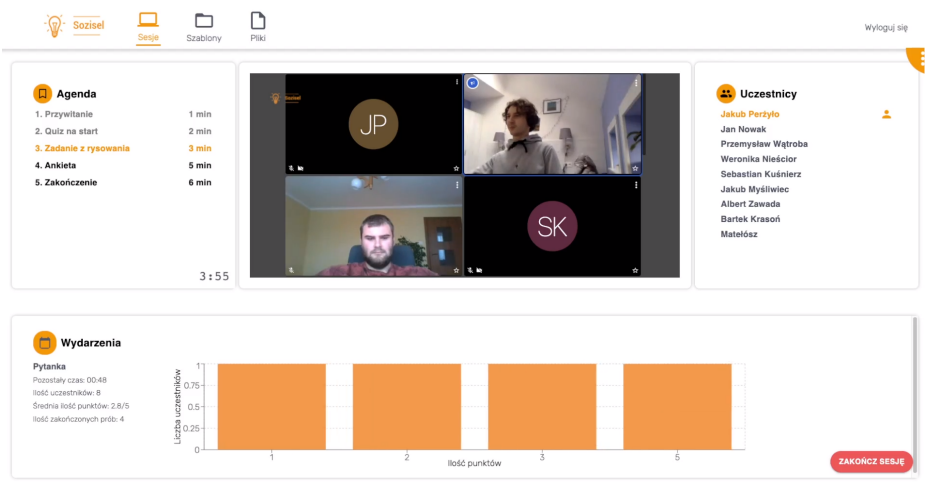

Fig. 9. Teacher view (just after a quiz was finished).

5 Conclusions and future work

Sozisel prototype proved to be a useful tool for medium-size online classes. It is based on open technologies, and designed in a modular way. The first direction of its further development is to leverage its modularity to incorporate more tools into the system or to integrate it with external services. Short-term goals include integrating game-based quizzes (developed at AGH university in another project), and computer networking specific services (we plan to use Kathara), and using it in real-world scenarios, for remotely led IT classes. As a long-term goal, we want to expose sozisel as a platform for EEDS, a model-based deployment environment for educational services, developed earlier at AGH [2]. In that way, multi-platform model-based automatic orchestration functionality would be added to the scheduling phase.

References

1. Feisel, L.D., Rosa, A.J.: The role of the laboratory in undergraduate engineering education. Journal of engineering Education **94**(1), 121–130 (2005)
2. Gandia, R.L., Zielinski, S., Konieczny, M.: Model-based approach to automated provisioning of collaborative educational services. In: Paszynski, M., Kranzlmüller, D., Krzhizhanovskaya, V.V., Dongarra, J.J., Sloot, P.M.A. (eds.) Computational Science - ICCS 2021 - 21st International Conference, Krakow, Poland, June 16-18, 2021, Proceedings, Part VI. Lecture Notes in Computer Science, vol. 12747, pp. 640–653. Springer (2021). https://doi.org/10.1007/978-3-030-77980-1_48, `https://doi.org/10.1007/978-3-030-77980-1_48`
3. Lee, H.S., Kim, Y., Thomas, E.: Integrated Educational Project of Theoretical, Experimental, and Computational Analyses. In: ASEE Gulf-Southwest Section Annual Meeting 2018 Papers. American Society for Engineering Education (2019)

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_58

4. Lynch, T., Ghergulescu, I.: Review of virtual labs as the emerging technologies for teaching STEM subjects. In: INTED2017 Proc. 11th Int. Technol. Educ. Dev. Conf. 6-8 March Valencia Spain. pp. 6082–6091 (2017)
5. Morales-Menendez, R., Ramírez-Mendoza, R.A., et al.: Virtual/remote labs for automation teaching: A cost effective approach. IFAC-PapersOnLine **52**(9), 266–271 (2019)
6. Perales, M., Pedraza, L., Moreno-Ger, P.: Work-in-progress: Improving online higher education with virtual and remote labs. In: 2019 IEEE Global Engineering Education Conference (EDUCON). pp. 1136–1139. IEEE (2019)
7. Zielinski, K., Czekierda, L., Malawski, F., Stras, R., Zielinski, S.: Recognizing value of educational collaboration between high schools and universities facilitated by modern ICT. J. Comput. Assist. Learn. **33**(6), 633–648 (2017). https://doi.org/10.1111/jcal.12207, `https://doi.org/10.1111/jcal.12207`

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_58