Information System for Meetings, Conferences and Trainings with Voting Possibility

Oleksii Bilyk and Iryna Vergunova

Taras Shevchenko National University of Kiev, 64/13, Volodymyrska Street, Kyiv, 01601, Ukraine

Abstract

The paper presents an automated system for conducting and documenting meetings of various kinds. A special feature of the developed system is the possibility of voting among the participants of the meetings and the generation of protocols based on the results of voting.

The developed system is a closed online platform. Users of this system receive some roles that can be changed as needed. The user subscribes to the channel of the relevant meeting and can see in real time the changes made by other users and make their own adjustments. The option "Voting" was added to the questions and the voting process included in the life cycle of the consideration of the issue. After the meeting, the necessary protocols are formed automatically. The implementation of this system is presented on the example of the current work of the Academic Council of the Faculty of Computer Science and Cybernetics.

Keywords [1]

Online platform, JWT Web Token, domain object, roles of participants, protocols, voting

1. Introduction

Recently, the world got acquainted with life in quarantine, with its beginning, people had to transfer professional activities online. The use of information systems with the ability to conduct video conferencing, document management and voting allows you to effectively perform work duties remotely, increases the comfort of creating and processing documents. Today, all developed countries are working on the development and implementation of such systems. Prominent examples are parliamentary electronic voting systems, corporate electronic document management systems, and automation systems for meetings and sittings. In 2020, Intecracy Group [1] presented a commercial solution for video conferencing with the possibility of voting. The system developed by this company is intended for use during meetings, which should result in legally significant actions. For example, meetings of the board of directors in business structures, discussion of regulations or decisions in the public sector, etc. Usually such meetings require a personal presence, which is a big problem in quarantine. Integration with the Megapolis.DocNet document management system is also possible [2]. However, all these systems are either automated voting systems or electronic document management systems. It should be noted that after the increase in the popularity of the Zoom service, the "Poll" functionality was added, in which you can put to the vote from one to several questions with the appropriate answer options, but the functionality is part of the video call. For example, after the transition to the online mode, the Academic council at the faculty of computer science and cybernetics used the Zoom video call service for meetings, and Google Forms for voting. Invitations to the meeting, agenda and invitations to vote were sent to council members by e-mail. Of course, in resolving the issue of communication and voting, the minutes of the counting commission were further processed manually, the meeting plan and all other documents for the work of the Academic Council were distributed as text documents, and invitations were sent to the council members by mail. After each meeting, the secretary manually formed the minutes of the meetings and the counting commission, as well as numerous extracts from the minutes.

Information Technology and Implementation (IT&I-2021), December 01–03, 2021, Kyiv, Ukraine
EMAIL: alexwhitecorp@gmail.com); vergunova@hotmail.com
ORCID: ; 0000-0003-3052-9143

2. Problem Statement

Ensuring that employees work effectively remotely in modern conditions is important. Remote meetings with support of various accompanying functions are also urgently needed. For this reason, the presented system was created, which allows to automatically notifying the participants about the meetings, to gradually form the minutes during the meeting, to vote with the results, etc.

Of course, some of the processes can be performed on different platforms and services (for example, sending invitations and notifications to meeting participants, forming minutes of meetings according to templates), which is not very convenient.

Therefore, the aim of our work was to create a system in which you can automatically perform the following processes: meeting planning; sending invitations and notifications to participants; formation of the agenda of meetings and submission of issues for consideration; voting of the present participants on specific issues with processing of results; formation of protocol cases according to a template (protocol of the counting commission, minutes of the meeting); management of roles and positions of participants; storage of general protocols and generated protocols of voting results.

The main tasks of the presented work are:

- to build an online platform that allows you to efficiently, conveniently and fully perform the preparation and conduct of meetings, to generate and to view appropriate protocols;
- to demonstrate the implementation of the system on the example of the current work of the Academic Council of the Faculty of Computer Science and Cybernetics.

Achieving the goal of the work leads to an online platform, which, unlike the existing ones, is at the same time a system for electronic document management, a system for preparing and holding meetings and a system for voting.

3. Technical solutions of basic problems

The first question was the selection of technologies that will be used to develop the platform. When considering the available variants, the emphasis was on reliability and productivity on the one side, as the meetings provide for real-time holding, and the possibility of easy expansion on the other one. As a result, it was decided to use the Spring Framework with its components (Spring Security, Spring MVC, Spring Data JPA, Spring WebSocket) to develop the backend side. On the frontend side it was decided to use the Vue.js framework, because, in the subjective opinion of the author, it is more convenient than React.js and lightweighter than Angular.js (although we are talking about platform development, Angular.js is too heavy in this situation). PostgreSQL is used as a database because it correspond the needs of the project.

When developing closed online platforms, especially those that provide documentation, secret ballot and division into different levels of access depending on the position and authority of users, the first issue is the choice of protocol and implementation of the authentication and authorization process. To develop the online platform "Academic Council" it was decided to abandon the standard approach with HttpSession, when to keep the user's status authorized in the system using cookie data and use the protocol authentication by tokens [3]. This method is most often used in the construction of distributed systems Single Sign-On (SSO), where one microservice (service provider or relying party) delegates the function of user authentication to another microservice (identity provider or authentication service). Thus, we immediately design the architecture of our code so that you can easily switch to a microservice architecture and perform horizontal scaling in case of successful project development.

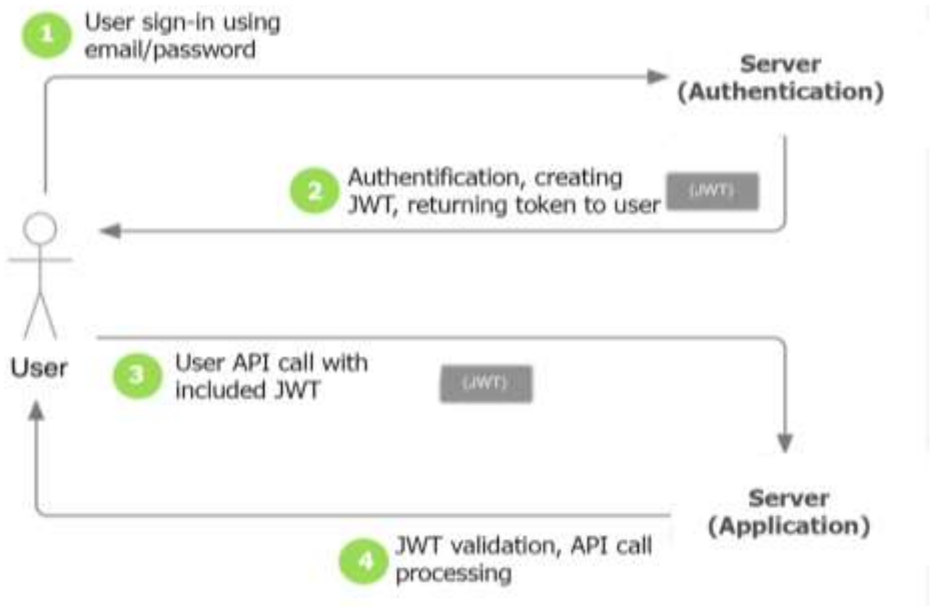
The implementation of this method is that the identity provider (IP) provides reliable information about the user in the form of a token, and the service provider (SP) application uses this token to identify, authenticate and authorize the user. There are several standards that define the protocol of interaction between clients and IP/SP applications and the format of tokens. Among the most popular standards are OAuth, OpenID Connect, SAML, and WS-Federation.

We chose the OAuth standard and the JWT Web Token format [4, 5]. The specified technology was used to verify user authentication according to the following algorithm (Figure):

1. The user makes the first authentication request using the "e-mail / password" pair.
2. The authentication service verifies the received data, in case of success creates a short-term Access Token [6] and a long-term Refresh Token [7] and sends them to the user.

3. When a user requests a platform API, he adds the previously received Access Token to it. When processing a request with an added access token, the server validates the received token and can authorize the user without additional requests to the database.

4. After the expiration time of the access token, the user requests the platform API, adding a Refresh Token to it. In response, the server creates a new access token.



1 User sign-in using email/password
Server (Authentication)
2 Authentification, creating JWT, returning token to user
(JWT)
User
3 User API call with included JWT
(JWT)
Server (Application)
4 JWT validation, API call processing

Figure 1: Algorithm of getting and using JWT token

To update the session with this platform, the user does not need to re-enter the pair e-mail / password. From the server point of view, this approach reduces the load on the database in the security layer, because you do not need to perform requests every time to verify the user and get general data about him – just check the validity of the token, and the required data is already encrypted. The issue of protection against compromised tokens remains open. It was solved as follows: to shorten the lifetime for access tokens, and in case of suspicion to remove the update token from the database, so the attacker immediately loses access to the platform. It should be noted that the platform is closed, so user can register only by receiving an invitation. The chairman of the Academic Council and the Secretary may invite a new participant. To do this, fill in the necessary information about the user, including his e-mail. After filling in all the fields to the specified mail finds an invitation letter with a link to register. Password recovery is also available. To do this, a link with a code is sent to the user's mail.

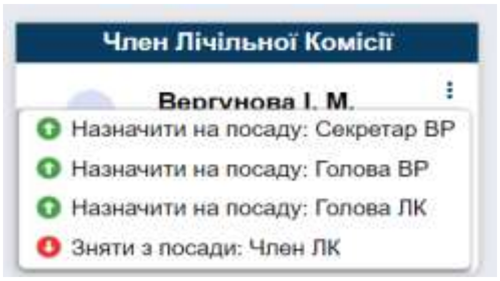
The next issue after authentication is the distribution of user roles by different levels of access and authority. The system uses the following roles for users:

- member of the Academic Council;
- member of the counting commission;
- member of the Academic Council;
- chairman of the Academic Council;
- secretary of the Academic Council;
- administrator.

In the specified list, the roles are listed in ascending order of access level. In this case, the secretary has more power, because the main function is moderation, which in a normal meeting and performs the secretary. The level of access determines the ability to use one or another functionality of the platform. For example, the minimum level of access to create a new meeting is the secretary of the academic council. You can view the current roles of the members of the Academic Council on the "Participants" page (Figure 2). You can also manage roles of users on this page using the appropriate drop-down menu, if available. If a user has been appointed or removed from office – he will receive a notification in the Notifications center (Figure 3).

The full list of meetings of the Academic Council is available for viewing on the “Meetings” page. You can also get brief information about the meeting, including the date and time, a link to the video conference and status. The meeting may have one of the statuses:

- moderation (completion of the necessary data and preliminary formation of the agenda; available for viewing only by the chairman and the secretary);
- planning;
- online (meeting now);
- completed.


Член Лічильної Комісії

Вергунова І. М.

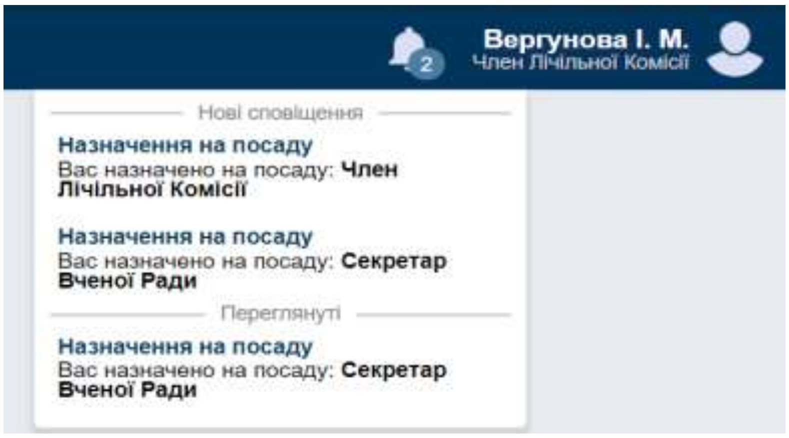
Назначити на посаду: Секретар ВР

Назначити на посаду: Голова ВР

Назначити на посаду: Голова ЛК

Зняти з посади: Член ЛК

Figure 2: Roles setting menu popup


Вергунова І. М.
Член Лічильної Комісії

Нові сповіщення

Назначення на посаду
Вас назначено на посаду: **Член Лічільної Комісії**

Назначення на посаду
Вас назначено на посаду: **Секретар Вченої Ради**

Переглянуті

Назначення на посаду
Вас назначено на посаду: **Секретар Вченої Ради**

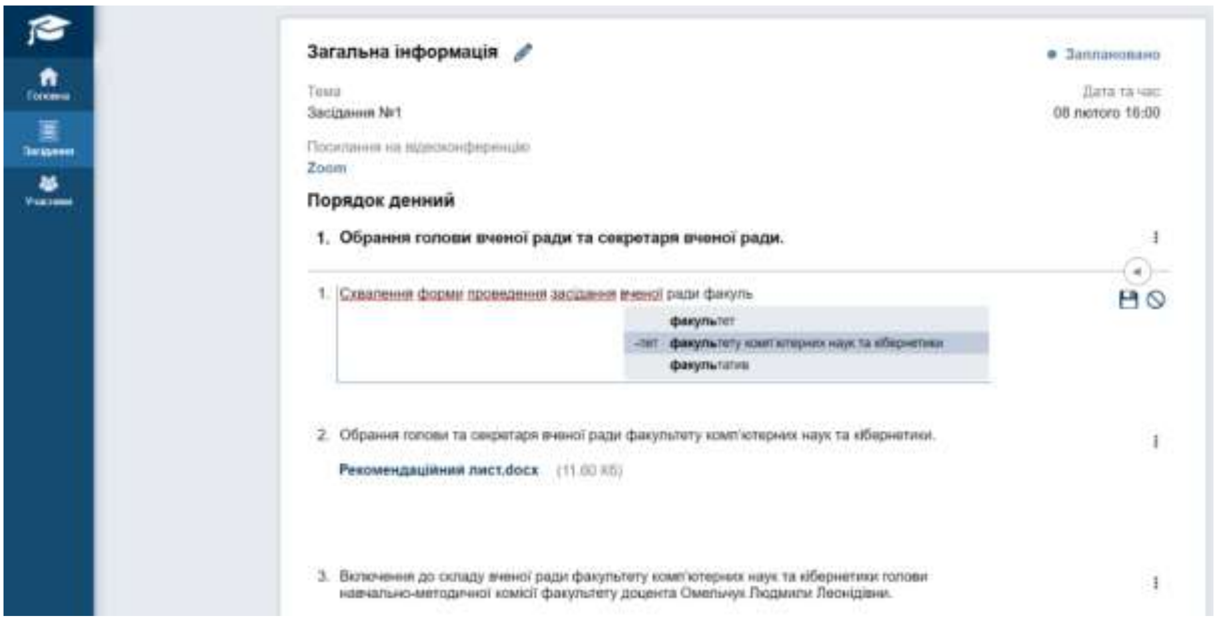
Figure 3: User notifications center

When you go to the meeting viewing page, the communication protocol with the server on WebSocket is changed [8-10]. The user subscribes to the channel of the relevant meeting and can see in real time the changes made by other users and make their own adjustments. He can edit general information on the meeting view page. Autocompletion is implemented to simplify the editing process (Figure 4). To do this, the minutes of meetings over the past years were analyzed and the most frequently used words and phrases were identified. They were structured and recorded and a special dictionary.

The words search in the dictionary is performed taking into account the beginning of the text that was entered, or through the use of abbreviations. Autocomplete information is also available from user profiles so you can quickly specify the rank and achievement of the desired meeting participant. Since in the Ukrainian language there is a concept of cases, all words and phrases are stored in the dictionary in all possible forms of declension. Also on the meeting page is the formation of the agenda: sections and issues are created for consideration. If necessary, you can add files (reports, presentations, statements, etc.) to each question, which can be downloaded by all users of the system. If the consideration of the created issue involves the voting of the members of the council, then the option "Voting" is added to the question and the voting process will be included in the life cycle of the consideration of the issue. This item was one of the key elements of the idea of creating this platform, as it is necessary to perform many standard movements to organize, conduct, obtain voting results, and then create the appropriate protocol in the desired format. It is also interesting that some votes must be secret, so it is not possible to simply keep what one or another participant of the meeting in an explicit form answered. You need to use a cryptographic approach.

Accordingly, the existing protocols of secret ballot were considered [11]. There were certain requirements for implementation. Our voting system must meet the standard requirements for secret ballot systems [11]. Namely, it requires that no one but the voter should know his choice, only registered participants can vote (they can vote only once) and the decision of the voter cannot be changed by anyone. In addition: each registered participant may be sure that his vote has been counted and can change his opinion and change his new choice over short time period; there is operator authentication;

we can find out who took part in the voting and who did not; the system maintenance should not require a lot of resources and it must be fault-tolerant in case of technical malfunctions, unintentional and criminal actions. As for the protocols available today, the simplest protocol of secret ballot is essentially a correspondence with electronic signatures between the election committee and a large number of voters. The first interesting protocol worth considering is the Two Agent Protocol. The basic idea is to replace one election agency with two so that they control each other.



Загальна інформація

● Заплановано

Тема
Засідання №1

Дата та час
08 лютого 16:00

Посилання на відеоконференцію
Zoom

Порядок денний

1. Обрання голови вченої ради та секретаря вченої ради.

1. Схвалення форми проведення засідання вченої ради факуль

факультет
–тет факультету комп'ютерних наук та кібернетики
факультатив

2. Обрання голови та секретаря вченої ради факультету комп'ютерних наук та кібернетики.

Рекомендаційний лист.docx (11.60 Кб)

3. Включення до складу вченої ради факультету комп'ютерних наук та кібернетики голови навчально-методичної комісії факультету доцента Омельчук Людмили Леонідівни.

Figure 4: Create and edit agenda items using autocomplete

Let's have a validator, whose duties include preparation of lists, and also admission or non-admission of the participant to voting. It creates a set of identification tags (exactly as many as registered participants) and sends one tag through the secure channel to each registered participant without information about who owns the tag. There are many different options based on the protocol of two agents. But with some complications these are:

1. Fujioka-Okamoto-Okhta protocol, slightly complicating the protocol, this solution uses masking encryption and partially solves the problem of collusion between the two agencies. It allows you to make sure that the document is genuine and signed by an authorized user, but does not let you know what data it contains.
2. Sensus protocol (as modification of the Fujioka-Okamoto-Ohta protocol). A characteristic feature here is that upon receipt of an encrypted message from the participant who voted, it is immediately added to the list of published results, and the signed ballot is additionally sent back to the participant as a receipt. This makes it possible not to wait until all other participants have voted and to end the voting (at the same time it becomes additional proof that a particular participant took part in the voting).
3. He-Su protocol. In it the key of the registered participant is signed with the blind signature (but the specified hash function is used and the method of blind encryption must be stipulated in advance). It also allows voters to change their minds by the end of the vote and further eliminates the possibility of collusion between the registrar and the agency.

The second approach is now chosen in our voting system. However, in the future, when expanding the functions of the platform, the issue of complicating the protocol algorithm will be considered in order to fully comply with the above requirements for secret ballot systems.

4. Results and Discussion

As already mentioned, as an example of the implementation of an online platform for meetings with the possibility of voting, the current work of the Academic Council of the Faculty of Computer Science and Cybernetics was considered. The developed system is closed. At the login stage, the user's authentication is verified using JWT technology with the involvement of the "e-mail/password" pair.

The system uses different user roles, all roles are displayed on the "Members" page. Roles have different levels of authority in using the functionality of the platform. For example, the minimum level of access to create a meeting of the Academic Council – the secretary. On the Participants page, the secretary can also manage the roles of the participants. If a user's role has changed, they will receive a notification in the notification center. On the "Meetings" page, after the secretary has entered the necessary information, a full list of scheduled and held meetings is available for all participants with brief information about each meeting, including the date and time, video conference link, status.

At the time indicated as the beginning of the meeting, the secretary or chairman of the Academic Council begins the meeting and the status changes to "online". After the change of status, the registration of council members for the meeting is activated. Registration occurs automatically when you visit the agenda page (Figure 5). The received information about the registered participants also used during the formation of protocols and voting.

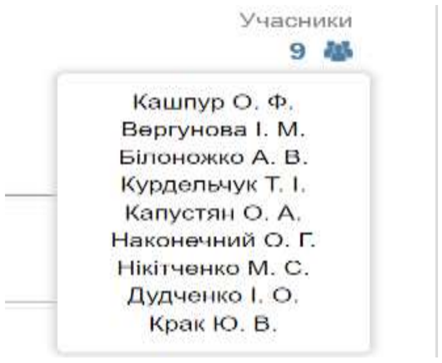

Учасники
9
Кашпур О. Ф.
Вергунова І. М.
Білоножко А. В.
Курдельчук Т. І.
Капустян О. А.
Наконечний О. Г.
Нікітченко М. С.
Дудченко І. О.
Крак Ю. В.

Figure 5: List of registered participants of the meeting

From the point of view of technical implementation, each meeting is a domain object, which is stored in the database and is a structure of three levels (the state diagram at Figure 6).

The first level is the Conference model, which contains general information about the meeting, as well as its related sections (next level) ConferenceChapter. The latter, by analogy, includes the very issues of the ConferenceQuestion meeting. The question model already retains all the textual information, and may also include voting, or additionally downloading files that are needed to address the question. It is from this model is obtained basically all the necessary information for the formation of the protocol. When we start to consider a question, a corresponding icon appears indicating the active question and the active section. The secretary also adds fields with autocomplete, which must be filled in for the protocol. The issues already considered are marked with a green check mark. In matters with the voting option, all council members have dynamic information about the voting process and its results. If the necessary information has already been heard on a specific issue and the secretary has started the voting process, each member of the council automatically opens a voting window in which he must vote. After the meeting, the necessary protocols are formed. It should be one file with the meeting protocols, which contains general information on the template (as well as a list of sections and issues discussed during the meeting). All questions in protocol are usually accompanied by textual recordings of the speeches and the results approved.

If a vote was taken to consider the issue, you must add information about the results of the vote in the specified format. When generating such a protocol, all data is automatically pulled from the agenda of the meeting, which does not require any additional manual processing. For each question, which provided for voting, a separate file is formed – the protocol of the counting commission with the appropriate serial number. All files formatted in .docx format. The received minutes can be downloaded and viewed by each user of the platform. We have: the meeting protocol is at the bottom of the agenda; the counting commission protocols are under the voting results of the relevant issue. If necessary, the secretary can download the protocol generated by the template, correct it and upload a new version of the file to the platform. Relevant events was recorded in the system. To form files on a template requires the use of a specific template engine [12]. In the case of working with .doc files, this may be a solution from Aspose [13], an example of the syntax:

```
<< [s.getName ()] >> says: "<< [s.getMessage ()] >>."
```

An important advantage of the approach is that the template is written directly to a .doc file. Because the syntax of the template is quite simple, we can easily adjust the template if you need to change the rules for filling out the protocol. However, this solution does not provide the necessary flexibility in editing file-filling styles, which does not allow you to play the desired template. We can also try to first generate the desired .html file using the Thymeleaf template [14]. This approach allows providing the necessary format and stylization of protocols due to the flexibility of the specified template.

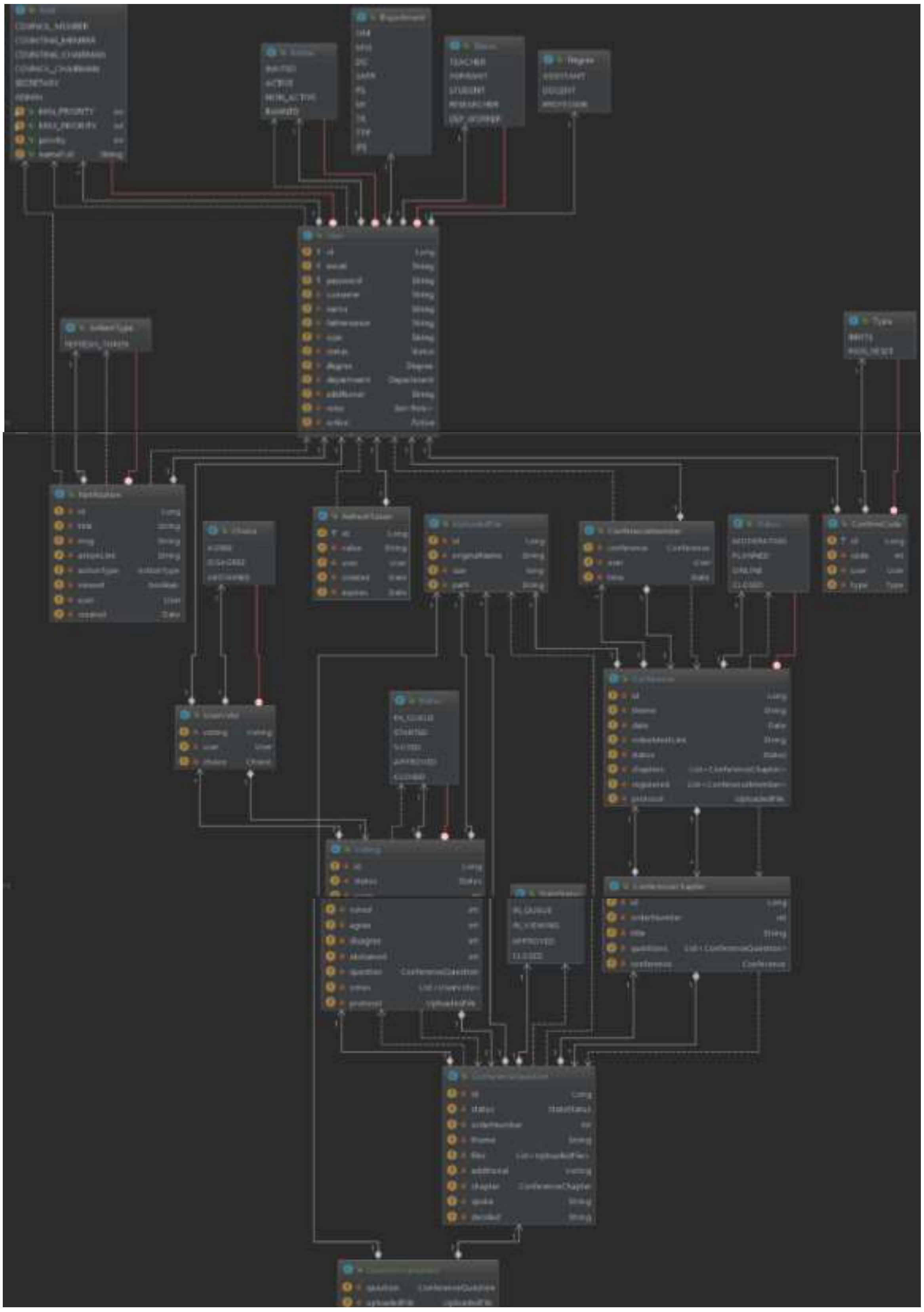
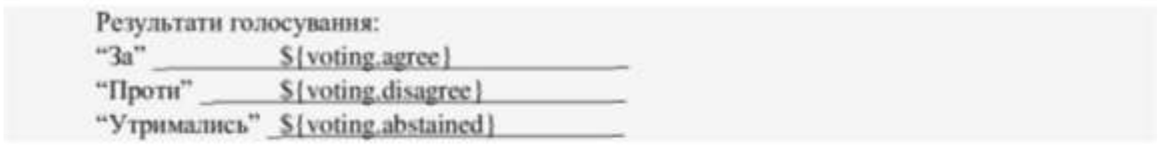


Figure 6: Domain model UML diagram

However, after conversion there are problems with stylization and the created file needs additional processing, which is not allowed by our condition of full automation of the process. Therefore, the system has its own template with the minimum required functionality. The syntax approach is borrowed from the already known Thymeleaf [14]. As in the first case, the code of the template is written to a .doc file, but immediately with the desired stylization, which is saved during processing by the template. Data (which are dynamic data) written by the algorithm to obtain the necessary data in the syntax of the created template. To do this, when processing the file through the template engine, a context is created. This context houses the objects of the domain model. We can get the necessary information from these objects. Thus, the created template in its syntax supports the use of variables by their names (Figure 7). It also supports the use of objects, calling their fields and methods, and therefore has the ability to use basic programming language classes to process data directly in the template, not in the code.



```
Результати голосування:
"За" ____________ ${voting.agree}______________
"Проти" ________ ${voting.disagree}____________
"Утримались" _${voting.abstained}_____________
```

Figure 7: The syntax example

In the case of using such a system, it is possible to abandon the usual way of saving protocols in .doc format. The protocols will be viewed immediately on the platform page, this will facilitate the work with extracts and their creation. If we need to use the protocol as an application, it will be enough to attach a link to the appropriate page of the platform, and not the whole file. We may also need to consider a system of plug-ins, because although meetings are mostly based on a single template, there are times that require a personal approach and implementation for each type.

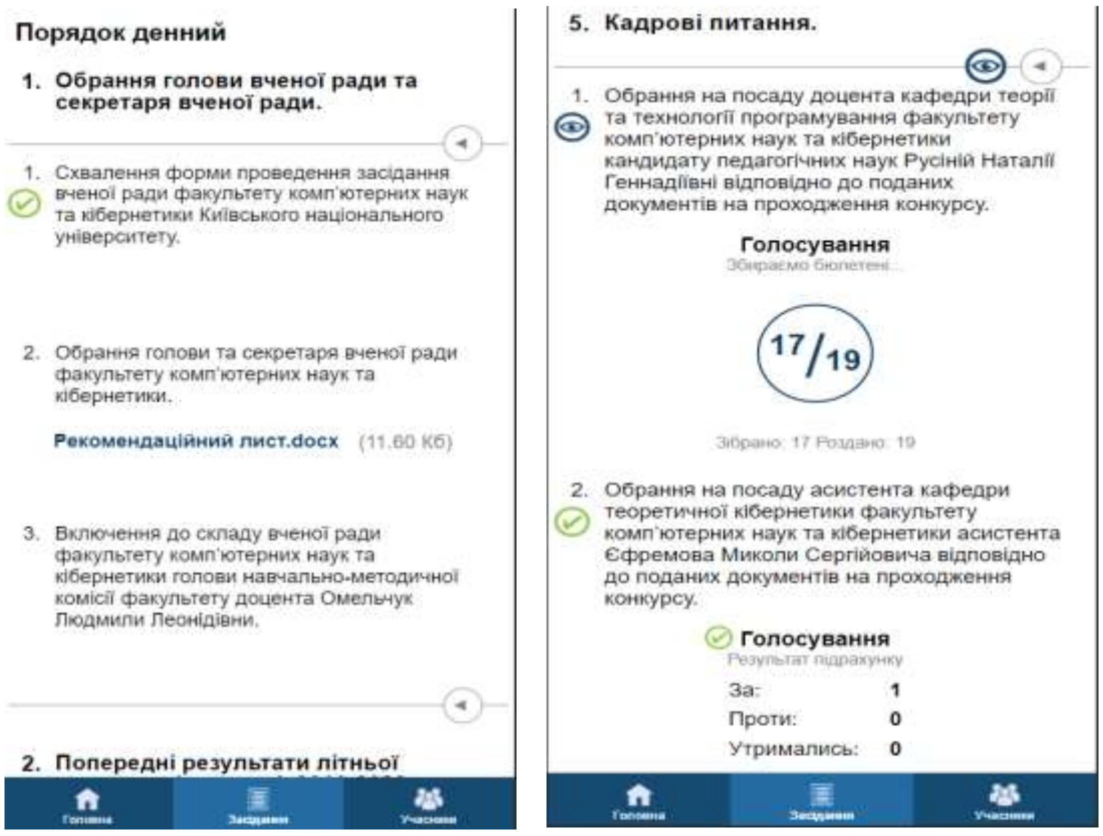
All registered participants of the meeting (except, of course, the secretary) can use smartphones to participate in the meeting and to vote. For this purpose, the mobile version of the user interface was implemented (Figure 8). The presented platform is now configured for holding meetings of scientific councils with automation of current work processes. Simple expansion of the system allows implementing it for use by scientific and labor collectives for carrying out of planned meetings of labor collectives, conferences with any number of working sections, training courses. For any use, the system provides a simple solution to the problem of holding online meetings, allows each participant to participate in the meeting only have a smartphone.

5. Conclusion

As a result of the research, we analyzed the algorithms of the current work of the academic council of the faculty and its meetings and selected the processes to be automated. We have developed algorithms for automated execution of key stages of organization and holding of meetings. Based on these algorithms, we developed a web application, demonstrated in this paper.

The peculiarity of the developed system is the convenience of preparation and holding of meetings, the possibility of secret ballot of registered participants with instant reflection of the voting process, automatic formation of protocols on the results of voting and the general minutes of the meeting.

The developed platform for holding meetings of the Academic Council with automation of current work processes and the system of secret ballot can be implemented both at other faculties and at the level of the institution. With a slight expansion, the system can be used for meetings of the staff, conferences with many sections, training courses. The resulting product solves the problem of holding online meetings, but is also adapted for the offline version during meetings in the hall. In this case, each board member only needs a smartphone to participate in the meeting. Also, after the implementation of the platform, it was noticed that the character of peak loads is present. For example, most meetings of the Academic Councils are held on Mondays, at other times it may be simple visits to common pages, work on the agenda or documents by several people, which does not create a high workload. Based on these observations, we consider the possibility of switching the platform to the Serverless model (in this situation we do not have a web server instance all the time, we only register lambda functions that work as web controllers; in this case we pay not for the time of use, but for the number of requests made). This will require dividing the platform into two functional parts: holding a real-time meeting (one service that requires a permanent instance); another platform functionality (working with documents, changing user roles, etc.), which is a request-response model and can be ported to Serverless.



Порядок денний

1. Обрання голови вченої ради та секретаря вченої ради.

1. Схвалення форми проведення засідання вченої ради факультету комп'ютерних наук та кібернетики Київського національного університету.

2. Обрання голови та секретаря вченої ради факультету комп'ютерних наук та кібернетики.

Рекомендаційний лист.docx (11.60 Кб)

3. Включення до складу вченої ради факультету комп'ютерних наук та кібернетики голови навчально-методичної комісії факультету доцента Омельчук Людмили Леонідівни.

2. Попередні результати літньої

Головна Засідання Учасники

5. Кадрові питання.

1. Обрання на посаду доцента кафедри теорії та технології програмування факультету комп'ютерних наук та кібернетики кандидату педагогічних наук Русіній Наталії Геннадіївні відповідно до поданих документів на проходження конкурсу.

Голосування
Збираємо бюлетені...

17/19

Зібрано: 17 Роздано: 19

2. Обрання на посаду асистента кафедри теоретичної кібернетики факультету комп'ютерних наук та кібернетики асистента Єфремова Миколи Сергійовича відповідно до поданих документів на проходження конкурсу.

Голосування
Результат підрахунку

За:	1
Проти:	0
Утримались:	0

Головна Засідання Учасники

Figure 8: Mobile version of platform for participants

6. References

[1] Intecracy Group presents the latest video meeting system, 2020. URL: https://softline.org.ua/news/intecracy-group-prezentuie-novitniu-systemu-videozasidan.html.

[2] Megapolis.DotNet electronic document management system, 2021. URL: https://in-base.com.ua/ua/soft/megapolis-docnet.html.

[3] Token Based Authentication Made Easy. Learn about token based authentication and how to easily implement JWT in your applications, 2016. URL: https://auth0.com/learn/token-based-authentication-made-easy/.

[4] Introduction to JSON Web Tokens, 2020. URL: https://jwt.io/introduction.

[5] S.E. Peyrott, JWT Handbook, Auth0 Inc., Version 0.14.1, 2016-2018. URL: https://assets.ctfassets.net/2ntc334xpx65/o5J4X472PQUI4ai6cAcqg/13a2611de03b2c8edbd09c3ca14ae86b/jwt-handbook-v0_14_1.pdf.

[6] OAuth official documentation. Access Token. 2021. URL: https://auth0.com/docs/tokens/access-tokens.

[7] OAuth official documentation. Refresh Token, 2021. URL: https://auth0.com/docs/tokens/refresh-tokens.

[8] I. Fette, A. Melnikov, The WebSocket Protocol, Internet Engineering Task Force (IETF), 2011. URL: https://datatracker.ietf.org/doc/html/rfc6455.

[9] WebSocket, in: Sovremenny`j uchebnik JavaScript, 2020. URL: https://learn.javascript.ru/websocket. (In Russian).

[10] WebSocket Protocol. SAP Help Portal. URL: https://help.sap.com/viewer/05d041d3df1a4595a3c45f57c15e2325/7.40.17/en-US/bd59428a3dd9454ab789412735adcb57.html.

[11] D.R. Patel, Information security: Theory and Practice, PHI Learning, Pvt. Ltd., 2008. Link.

[12] Java Template Engines, Topstone Software Consulting, 2018. URL: http://topstonesoftware.com/publications/java_template_engines.html.

[13] Aspose Template Engine demo, 2020. URL: blog.aspose.com/2020/01/14/generate-word-documents-from-templates-dynamically-using-java/.

[14] Thymeleaf official documentation. URL: https://www.thymeleaf.org/documentation.html.

[15] Docx4j – Getting Started, Plutext Pty Ltd, 2021. URL: https://raw.githubusercontent.com/plutext/docx4j/master/docs/Docx4j_GettingStarted.pdf.