CIVIL ENGINEERING JOURNAL

Article no. 41

THE CIVIL ENGINEERING JOURNAL 4-2022

DEVELOPING A VIRTUAL OPEN-AIR MUSEUM OF VERNACULAR ARCHITECTURE

Tomáš Bouček[1], Martin Landa[1] and Petr Soukup[1]

Czech Technical University of Prague, Faculty of Civil Engineering, Department of Geomatics, Prague, Thakurova 7/2077, Czech Republic; tomas.boucek@fsv.cvut.cz

ABSTRACT

Vernacular architecture is an integral part of the national cultural heritage. Today, however, many of these buildings exist only on old plans or photographs and the average citizen has no opportunity to get acquainted with this part of the national identity. Therefore, in our work, we present the development of two web applications with the aim of creating a virtual open-air museum for presenting vernacular architecture in the Czech Republic. The applications were created using open-source technologies, and are implemented with methods that allow easy transfer from one operating system to another. The presented content is a carefully selected sub-sample of more than 10,000 available records representing all regional types of vernacular architecture. The result is one application designed for editors to manage the presented content and one application allowing interactive viewing of the available geo-located records designed for the general public. Individual records can be searched either directly using the map window or by querying the attribute table. These records contain descriptive information about the object, as well as historical photographs and plans and, for some objects, additional information in the form of 3D models, PDF documents and other files. The applications are designed in such a way that their content can be freely expanded in the future and thus contribute to the popularization of vernacular architecture among the general public, which was the main reason for their creation.

KEYWORDS

Web application, Deployment, Docker, Map application, Open-air museum, Open source, QGIS, Gisquick, Vernacular architecture, 3D model, Component-Based Software Engineering, Dynamic architecture, Web-based DBMS

INTRODUCTION

The use of open-source software technologies is increasingly popular across almost all fields related to the development of software tools, web application development is not an exception. Therefore, in this paper, we present two web applications that were created using a combination of several open-source software packages. Both applications were developed within the VISKALIA (Virtual open-air museum of vernacular architecture) project, which is supported by the Ministry of Culture of the Czech Republic [1].

The main objective of the VISKALIA project is to document and present the vernacular architecture in the Czech Republic. Vernacular architecture can be considered an important part of regional and national identity. In the collections of the National Museum of the Czech Republic and the Institute of Ethnology of the Czech Academy of Sciences, there is a large number of historical documents from this field (plans, photographs, drawings, textual descriptions) in several separate collections. Within the framework of the project, these documents were digitized, stored in a unified manner in a database (hereinafter referred to as a central database) and made available to the public on the web. The central database contains more than 10,000 records and its web interface allows users to search for records according to several basic criteria (e.g. location, regional type, purpose, object material).

The comprehensive set of records in the central database is a unique source of information, especially for the professional public. The general public finds it more difficult to navigate such a large and structured database. Therefore, the virtual museum VISKALIA was created to present the basic features of the cultural heritage contained in vernacular architecture in a clearer way. It contains approximately 200 carefully selected buildings (objects) representing all regional types of vernacular architecture in the Czech Republic. All the documents (records) registered for these objects in the central database are also available in the virtual open-air museum in the form of a database (hereinafter referred to as a selective database). In addition, other specific information such as text documents or photographs, as well as spatial models, can be recorded for the objects in the selective database.

Spatial models have a great potential in terms of visualization and attractiveness for users. However, the creation of detailed models of building objects is a very laborious task, complicated in our case by the fact that the selected historical objects usually no longer exist, or their original (modelled) state has been changed by reconstruction. It is therefore necessary to rely on the surviving documents, which are most often plan documentation and period photographs or drawings. Colleagues from the National Museum and the Institute of Ethnology selected a total of 15 buildings (Appendix Table A1), which represent all regional types of rural buildings. Details of the creation of models from plan documentation are described in a separate article [2].

Physical models of buildings, which have already been created for some objects in the past and are stored in the archives of the National Museum, can also be a specific type of basis for digital modelling. For this purpose, a total of three objects were selected (Appendix Table A2), one of which contains an interior that was also the subject of digitisation.

The creation of a web application with a focus on vernacular architecture is also discussed by Ammendola et al. [3], mentioning that this way of disseminating information is the best option to reach the attention of the general public. Another advantage mentioned is the possibility of constant improvement and integration of other records. Other authors also mention the importance of sharing cultural heritage. Dhonju et al. [4] describe the creation of an online geo-crowdsourcing system designed for large-scale documentation and sharing of cultural heritage. One of the authors' reasons for creating such a system was the fact that cultural heritage around the world is under considerable risk from natural and anthropogenic threats. The creation of a web application for cultural heritage has also been addressed by Mantegari et al. [5] and Chalkidou et al. [6]. An important part of documenting cultural heritage is the creation of 3D models and their subsequent presentation. Therefore, Guarnieri et al. [7] deal in their work with a case study regarding the development of a web-based application for the interactive exploration of 3D models. Nishanbaev [8] also deals with 3D models of cultural heritage in his work. Specifically, his goal was to create a web repository for 3D models. Recently, augmented and virtual reality has become increasingly popular. Their usage in the presentation and preservation of cultural heritage has been addressed in the works of Rattanarungrot et al. [9] and Selmanović et al. [10] respectively.

Another example of web application development is described in the work of Jílková & Cajthaml [11], who worked on the creation of a database of historical atlases in the Czech Republic. Web application with similar content has already been created in another project called "A Transformation of Rural Architecture with Emphasis on the Development of the 19th and 20th Centuries" [12]. Specifically, this is a map application (available from https://country.fsv.cvut.cz/) that has been discussed as a possible basis for our map application, although the technologies used in its creation are not exactly state-of-the-art nowadays (framework based on PHP). The development of the application was preceded by the creation of the web database described in the article by Soukup & Sýkora [13]. The second option was to start with the map application from scratch, which we eventually chose. Our inspiration was a map application dealing with hydrological data (available from https://rain.fsv.cvut.cz/webapp/gisquick/), which partly uses the same open-source technologies as in our case, as described by Landa et al. [14].

Based on the topic that our web applications address and the requirements that the applications should meet, we set the overall goal divided into the following objectives:

- attractive presentation of rural architecture in virtual space;
- separate user interfaces for editors and visitors;
- easy deployment of the whole system.

METHODS

A high-level static architecture of our solution was designed based on system requirements and the objectives highlighted in the introduction section. The system design comes from Component-Based Software Engineering (CBSE) [15]. Based on the static architecture of the system, a dynamic architecture was developed. It allows the evaluation of the interaction between system components in a sequential order. Workflow diagrams were designed in the Unified Modeling Language (UML). The system was programmed using Bash, Python, Structured Query Language (SQL), and JavaScript. Web applications were designed using Django and Gisquick frameworks. Spatial models were visualized using Three.js library [16]. For system deployment Docker technology [17] was used.

RESULTS

System design

This section describes the system design. A high-level static architecture presents the core software components of the system. A system dynamic architecture shows interaction between the core components.

High-level system architecture

The high-level system architecture is based on a client-server model. System software components (Table 1, Figure 1) are decomposed into three layers: persistence, application and client layer the persistence layer provides selective (P1) and publication (P2) databases. Software components in this layer preserve the data for further processing and publication. The application layer contains a deployment package (A1) which controls system deployment of two web applications and related databases. The first web application is dedicated to the editors (A2). The application provides access to the data stored in the selective database (P1). The second web application (A3) presents data from the publication database (P2), which is deployed from the selective database (P1) by A1 package. The functionality of web applications A2 and A3 is provided in the client layer via a web browser (C1).

The system design considers three types of users: system administrator, editor and visitor. The system administrator may easily deploy and manage the system. The editor may add or modify additional information served by the selective database (P1) using web application A2. The visitor may interactively browse data content provided by the publication database (P2) via map application A3.

Tab. 1 - System components overview.

ID	Component name	Layer	Description
P1	Selective DB	Persistence	Store selected data from central DB
P2	Publication DB	Persistence	Store geospatial data for publication
A1	Deployment package	Application	Deploy the system
A2	Editorial App	Application	Web application for editors
A3	Map App	Application	Web map application for the public
C1	Web browser	Client	Software component to access A2, A3

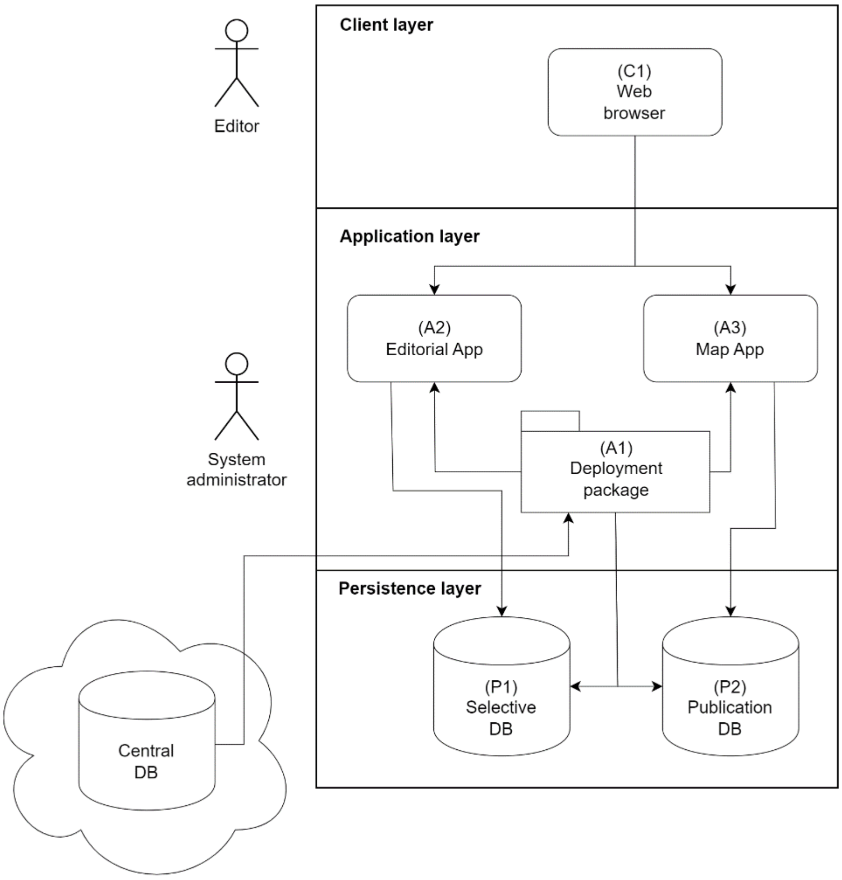

Fig. 1 – Static high-level system architecture.

Dynamic system architecture

The dynamic architecture design reflects the interactions of the system software components (Table 1) in time. The UML sequential diagram Figure 2 shows the interaction between the software components A1, A2, A3 (application layer) and P1, P2 (persistence layer). The deployment process starts with the initial configuration of the system. Next, the deployment package (A1) performs four major operations. First, the selective database (P1) is deployed. By this operation selected data records are imported from the central database into the selective database, the data is normalized and a synchronization process between central and selective databases is activated. It ensures that data provided by the selective database will always be synchronized with the central database content. Second, the editorial web application (A2) is deployed. Third, the publication database (P2) is deployed and geospatial features, required for the visualization of the objects in the interactive map, are created. Four, the web map application (A3) is deployed.

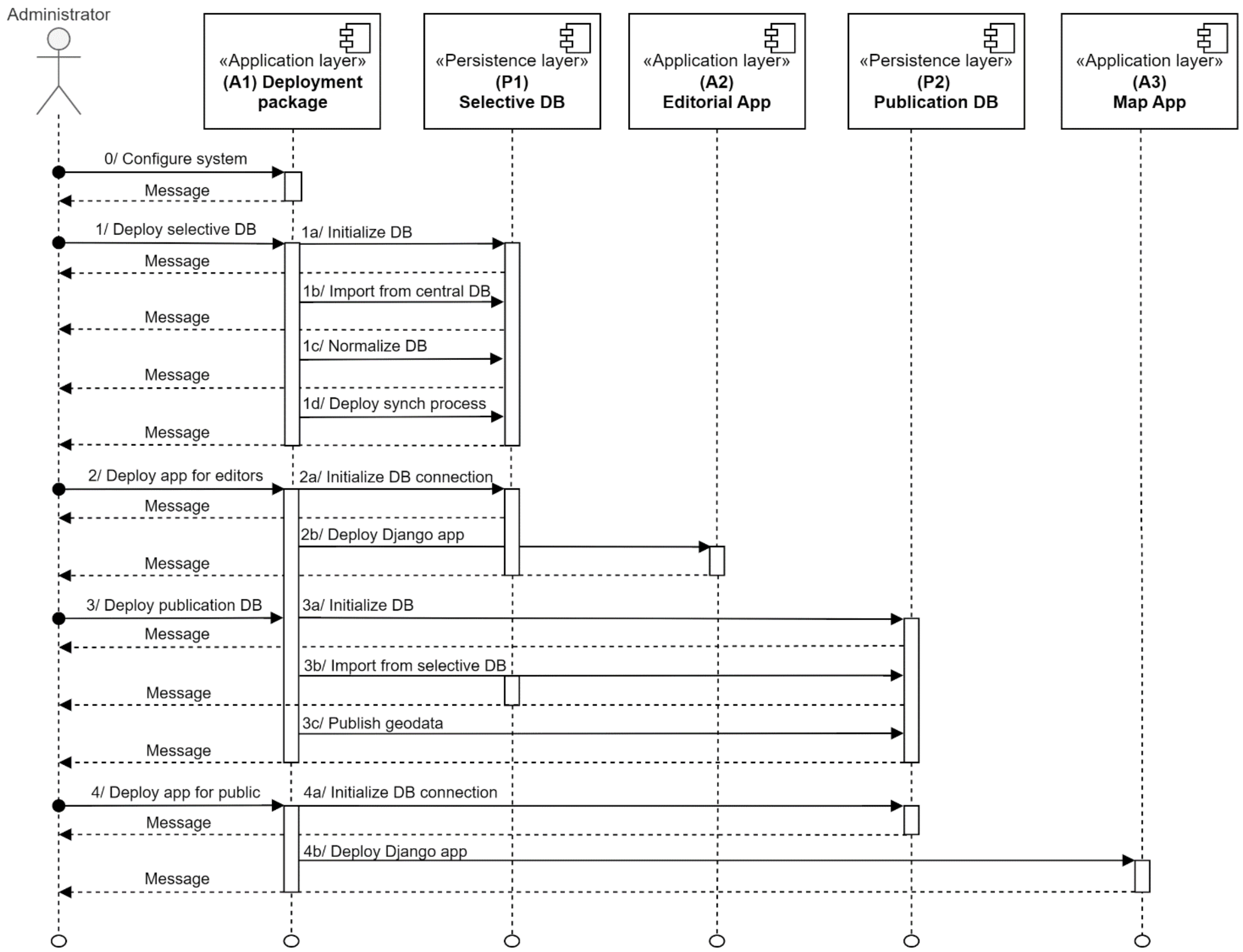

Fig. 2 – Dynamic system architecture.

Implementation

Two web applications were created - the editorial and the map application. Only open-source software components were used to achieve the desired outcomes. In the following sections, both applications including the deployment package and related databases are described in more detail in terms of development and functionality.

Deployment package

The deployment package was programmed in Bash, Python and SQL. The deployment process is controlled by Docker containers and a docker-compose tool. There are five deployed services (Docker containers):

- db: responsible for the selective database (P1);
- syncdb: responsible for synchronization between the selective (P1) and the central database;
- phpmyadmin: provides access to the selective database (P1) via web interface for system administrators;
- editapp: responsible for editorial web application deployment (A2);
- mapapp: responsible for web map application (A3) and the publication database (P2) deployment;

The deployment of the complete system is done by a command below:

```
$ docker-compose up
```

Selective and Publication databases

The selective database is provided by object-relational database management system MariaDB. The database contains three relations:

- objects_of_interest: the main relation describing objects of interest (buildings)
- files_central: information about files (photographs, plans) taken from the central database
- files_selective: files (spatial models, photographs, documents, etc.) managed by the selective database

Records from files_central and files_selective are linked to objects_of_interests via the foreign key defined by the OID attribute (Figure 3).

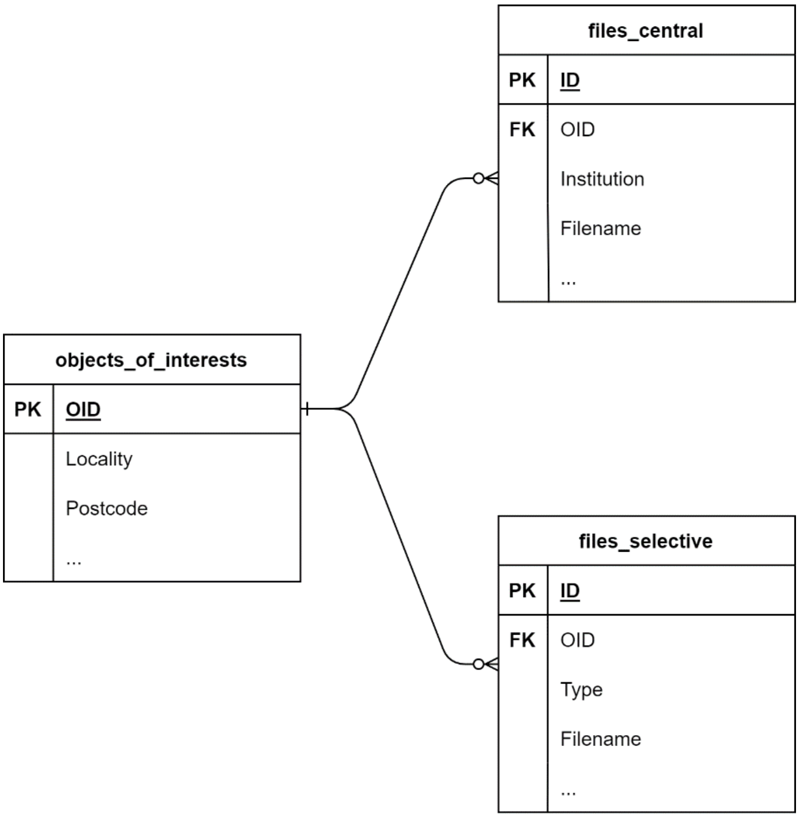

Fig. 3 – Selective database entity relationship diagram.

The publication database is deployed from the selective database supplemented by point geometry. The database is driven by SQLite file-based format extended by OGC Geopackage specification[1].

Editorial Web Application

To design the editorial application (Figure 4), the open-source high-level Python web framework Django[2] was used. The editorial application manages the selection of objects from the central database and access to it is authorized. It is therefore intended only for the creators and administrators of the selective database, which contains objects of interest of the VISKALIA project. The editors may view and edit individual records, which can be searched by keywords or filtered by district or region. It is then possible to add new files (photos, documents, models) to a specific record,

[1] OGC GeoPackage https://www.geopackage.org/
[2] Django project https://www.djangoproject.com/

which represents the content of the selective database. The editorial application is regularly synchronized with the central database to guarantee its up-to-dateness.

Fig. 4 – Editorial Web Application.

Web Map Application

Map application (Figure 5) is designed for the general public, so anyone can access it. It allows users to interactively view individual geographically localized records and display information about them using the attribute table or, better, a customized infopanel. Users can switch between base layers (base map, orthophoto and historical orthophoto) or display administrative boundaries (districts, regions or state border). It is also possible to filter between individual records using the already-mentioned attribute table. In addition, the application also allows to measure distances between interactively entered points and determines the size of manually defined areas.

The application generally uses the Czech localization, but the user interface can be switched to English. However, the information displayed for individual objects is taken from a database that is not managed by us, so the possibility of having this information in English in the future seems unfortunately unlikely.

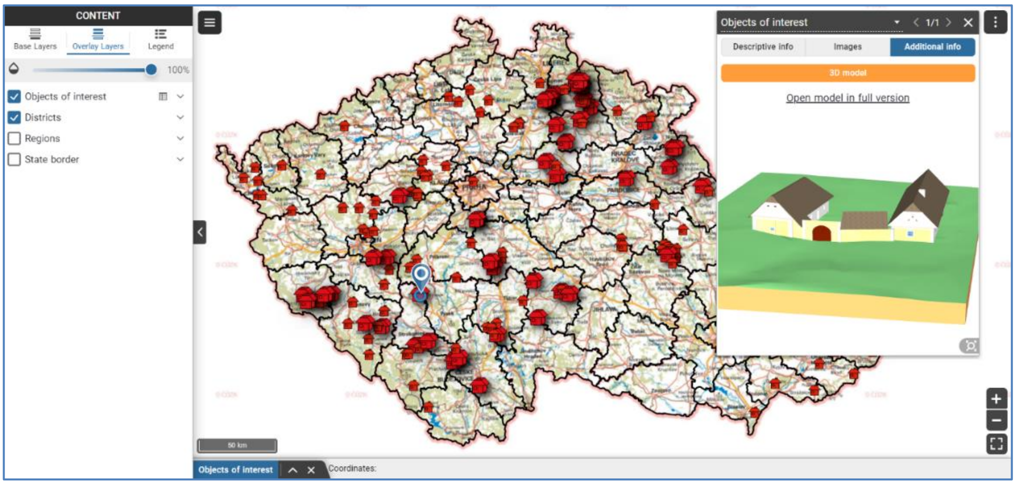

Fig. 5 – Web Map Application.

The combination of QGIS software[3], which was used to create a project with all desired layers (base layers, overlay layers and symbology), and Gisquick[4] publishing platform was used to create the map application. The individual published layers are summarized in Table 2. The Gisquick QGIS plugin was used to upload the project to the publication server. The server can be accessed using a web interface. Three layers were defined as the base and six layers as the overlays. For base layers, Web Map Tile Service (WMTS) was used to get the fastest possible data retrieval. Two of the six overlay layers are not viewable because they do not contain geometry, the layers contain additional information about objects like photographs, plans, spatial models, etc.

Tab. 2 - Overview of the published layers.

Layer	Type	Viewable	Queryable
Base map	Base layer	yes	no
Orthophoto	Base layer	yes	no
Historical orthophoto	Base layer	yes	no
Objects of interest	Overlay layer	yes	yes
Files from central DB	Overlay layer	no	yes
Files from selective DB	Overlay layer	no	yes
Districts	Overlay layer	yes	no
Regions	Overlay layer	yes	no
State border	Overlay layer	yes	no

Gisquick customization (info panel)

Information about a specific object can be displayed more clearly in the aforementioned infopanel, which is displayed in the right part of the map window after clicking on the object of interest. However, in its generic form, it displays information about the objects of interest in a similar way to the attribute table, which was not sufficient in our case, either from an informational or aesthetic point of view. Our goal was to display information about the object not only from the attribute table of the corresponding layer, but also from associated tables containing information about photos and other files of the object. Photographs and other files, such as spatial models or PDF documents, were also intended to be displayed in the infopanel. Therefore, it was necessary to customize the infopanel according to our needs. For the purpose of customization, including the implementation of the required functionality, the JavaScript framework Vue.js[5] was used. To upload the customized infopanel to the Gisquick server, it first had to be compiled into the necessary format. The npm command line utility was used for this compilation.

Unlike the generic infopanel, the customized infopanel consists of three parts. The first part of the infopanel relates to descriptive information such as location, description number, municipality, district, house type or a more specific description of the property. All the information displayed in the first tab is taken from the objects_of_interest relation (Figure 3). The second part is designed for displaying images (historical photographs, plans, etc.) of the object and related information. It allows clicking on the images to view them outside the infopanel in full size or save them to a local device. The files from the central database are the source of information for the second part of the infopanel (files_central relation in Figure 3). The third part is dedicated to additional files (files_selective relation in Figure 3) such as 3D models, PDF documents, additional images not included in the central database and possibly other files (e.g. DOCX, TXT, etc.). In the case that additional files are

[3] QGIS project https://www.qgis.org/
[4] Gisquick publication platform https://www.gisquick.org/
[5] Vue.JS project https://vuejs.org/

uploaded, the third section is further subdivided into one to four sub-sections. The content corresponds to the distribution of each possible file. The second and third parts of the infopanel are shown in Figure 6.

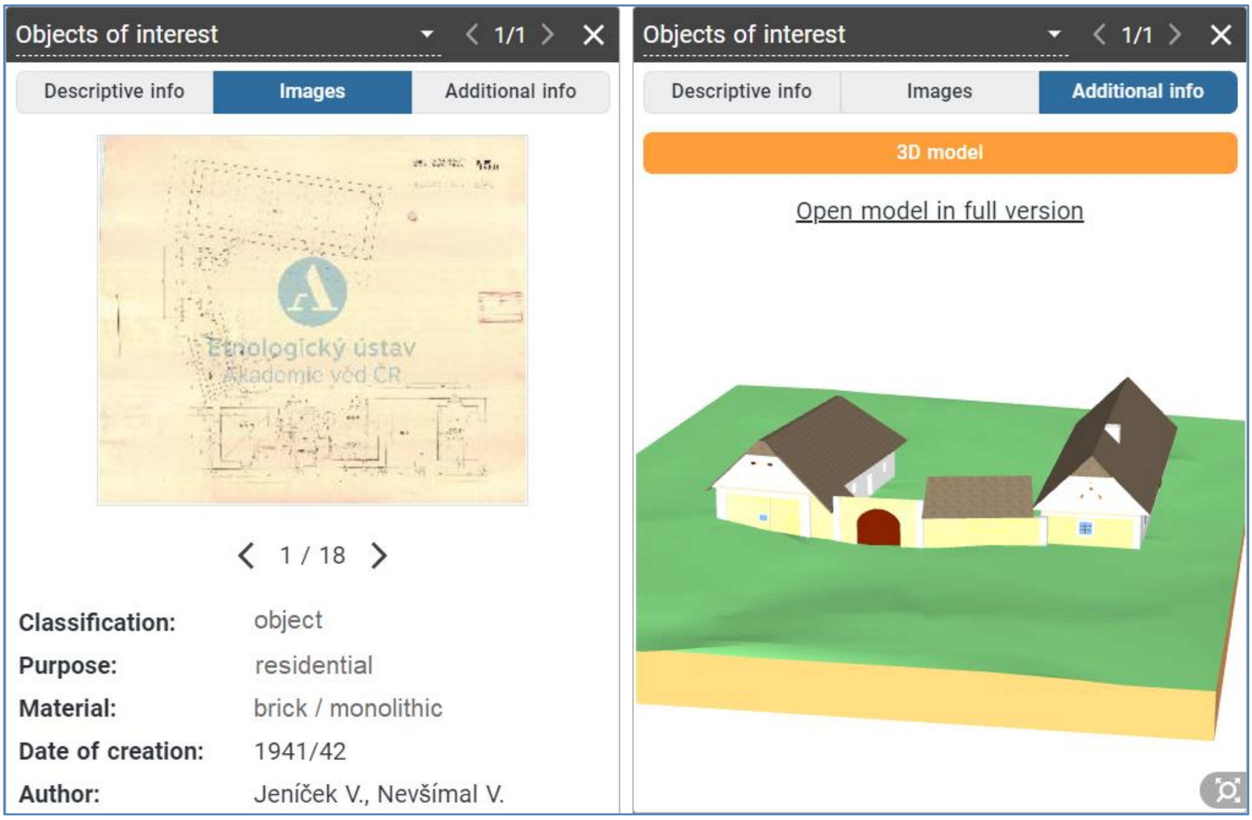

Fig. 6 – The second and the third part of infopanel.

In the preview, the 3D models are only in the basic version, but it is still possible to change the view or zoom in and out of the model. However, there is also a link to the full version of the 3D model providing additional functionality such as showing and hiding different layers of the model (Figure 7). PDF files are displayed in the infopanel similarly to images, where clicking on the preview opens a window with the full version of the PDF file. Any additional files only have a link in the infopanel to either download the file or to display it in a new window if the web browser is capable of doing so.

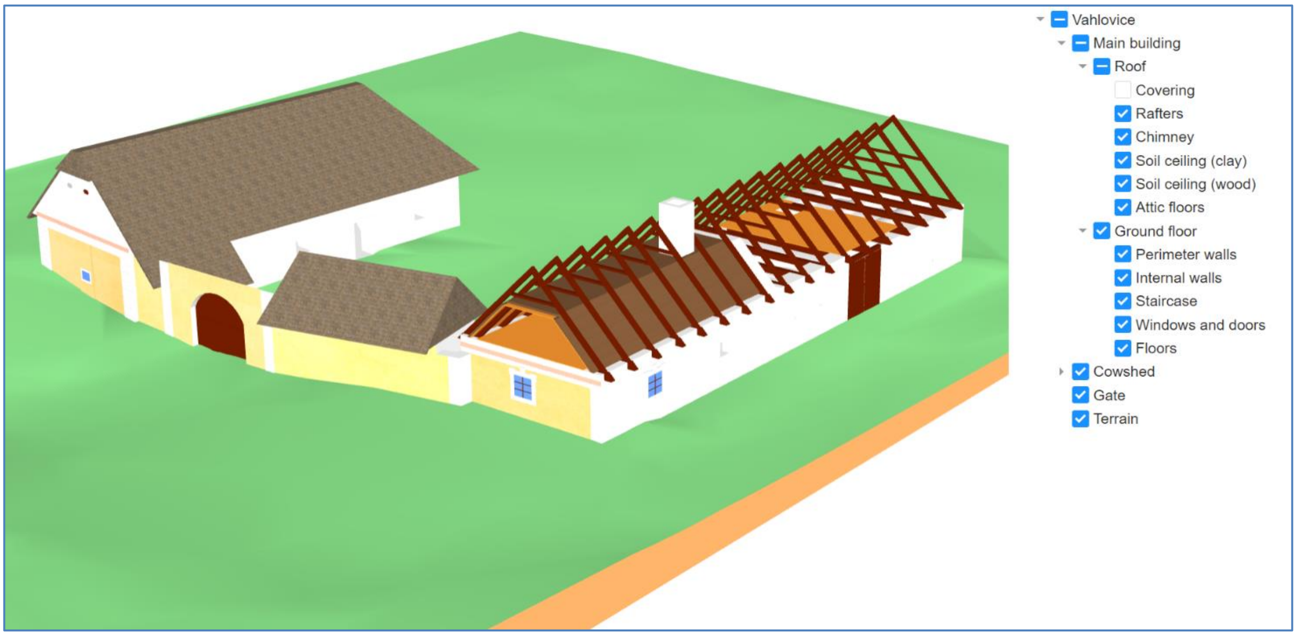

Fig. 7 – Interactive viewing of the 3D model by layers.

CONCLUSION

The article describes the creation of software for the implementation of a web-based open-air museum of vernacular architecture in the Czech Republic. The basis of the open-air museum called VISKALIA are two separate applications that provide all the necessary functions for the management and presentation of its content. The content management application (https://skansen.fsv.cvut.cz/editapp/) is intended for authorized editors who fill the selection database with guaranteed content. The application for the presentation of the content of the open-air museum (https://skansen.fsv.cvut.cz) is intended for all those interested in vernacular architecture, i.e. both the general public and experts.

At present, the virtual open-air museum contains general information on approximately 200 buildings, of which detailed spatial models have been created for 18 buildings. A total of 15 of these models were created on the basis of surviving plan documentation using CAD tools. The remaining three models were created by digitizing existing physical models using the photogrammetric method Structure-from-Motion.

The number of building objects integrated into the virtual open-air museum can be freely expanded according to the needs without the need to modify the software solution. Only open-source tools were used for the development, management and presentation of the open-air museum. The chosen basis for working with spatial data is the QGIS platform, extended by the Gisquick publishing platform. In order to present the content of the virtual museum in an attractive way, it was necessary to perform specific programming modifications to standardized tools, especially to the content of the integrated infopanel.

The whole system is modular and easily transferable to other technical equipment. While the development and testing of the web applications took place on the technical equipment of the Czech Technical University in Prague (the workplace of the system developers), the production version of the museum will be deployed on the infrastructure of the National Museum (the workplace of the system administrators). The easy portability of the applications and all the interconnected software is ensured by the chosen solution based on Docker containers.

The created virtual open-air museum VISKALIA presents selected information about important objects of rural architecture in a unified way. We assume that in the future the number of presented objects and information will gradually increase. We expect that the practical experience of the users of the virtual open-air museum will also provide valuable suggestions for its further development.

We believe that the created virtual open-air museum will contribute to the popularization of vernacular architecture among the general public and will become an interesting platform for the professional community.

ACKNOWLEDGEMENTS

This work was supported by the Ministry of Culture of the Czech Republic from the Program for the Support of Applied Research and Development of National and Cultural Identity for the Years 2016-2022 (NAKI II), grant project "VISKALIA – Virtual Open-air Museum of the Vernacular Architecture ", No. DG20P02OVV003.

REFERENCES

[1] VISKALIA project. Available online in Czech: http://viskalia.fsv.cvut.cz/projekt/index.html (accessed on 20 October 2022)

[2] Poloprutský Z., Frommeltová E., Münzberger J., & Sedlická K., 2022. 3D Digital Reconstruction of Defunct Rural Buildings Based on Archival Sources. The Civil Engineering Journal 31(1), 196–210, Prague. DOI: https://doi.org/10.14311/CEJ.2022.01.0015

[3] Ammendola J., Dipasquale L., Ferrari E. P., Mecca S., Montoni L., & Zambelli M, 2022. A collaborative Web App to foster a knowledge network on vernacular herit-age, craftspeople, and sustainability. DOI: https://doi.org/10.4995/HERITAGE2022.2022.14951

[4] Dhonju H. K., Xiao W., Mills J. P. & Sarhosis V., 2018. Share Our Cultural Heritage (SOCH): Worldwide 3D Heritage Reconstruction and Visualization via Web and Mobile GIS. ISPRS Int. J. Geo-Inf., 7, 360. DOI https://doi.org/10.3390/ijgi7090360

[5] Mantegari G., Palmonari M. & Vizzari G, 2010. Rapid prototyping a semantic web application for cultural heritage: the case of MANTIC. In Extended Semantic Web Conference (pp. 406-410). Springer, Berlin, Heidelberg. DOI: https://doi.org/10.1007/978-3-642-13489-0_33

[6] Chalkidou, S., Arvanitis, A., Patias, P. & Georgiadis, C., 2021. Spatially Enabled Web Application for Urban Cultural Heritage Monitoring and Metrics Reporting for the SDGs. Sustainability, 13(21), 12289. DOI: https://doi.org/10.3390/su132112289

[7] Guarnieri A., Pirotti F. & Vettore A., 2010. Cultural heritage interactive 3D models on the web: An approach using open source and free software. Journal of Cultural Heritage, 11(3), 350-353. DOI: https://doi.org/10.1016/j.culher.2009.11.011

[8] Nishanbaev I. (2020). A web repository for geo-located 3D digital cultural heritage models. Digital Applications in Archaeology and Cultural Heritage, 16, e00139. DOI: https://doi.org/10.1016/j.daach.2020.e00139

[9] Rattanarungrot S., White M., Patoli Z. & Pascu T., 2014. The application of augmented reality for reanimating cultural heritage. In international conference on virtual, augmented and mixed reality (pp. 85-95). Springer, Cham. DOI: https://doi.org/10.1007/978-3-319-07464-1_8

[10] Selmanović E., Rizvic S., Harvey C., Boskovic D., Hulusic V., Chahin M. & Sljivo, S, 2020. Improving accessibility to intangible cultural heritage preservation using virtual reality. Journal on Computing and Cultural Heritage (JOCCH), 13(2), 1-19. DOI: https://doi.org/10.1145/3377143

[11] Jílková P., Cajthaml J., 2019. Database of Historical Atlases: An Interactive Web Application. In: Proceedings of the International Cartographic Association. Tokyo: International Cartographic Association, 2019, vol. 2. ISSN 2570-2092. DOI: https://doi.org/10.5194/ica-proc-2-52-2019

[12] Proměna venkovské architektury [online]. Available online in Czech: http://venkov.fsv.cvut.cz/projekt/index.html (accessed on 20 October 2022)

[13] Soukup P., Sýkora K., 2020. Development of Rural Architecture – Design and Creation of a Web Database. The Civil Engineering Journal, 29(4), 490–499, Prague. paper: http://venkov.fsv.cvut.cz/projekt/publikace/2020_PS_KS_CEJ.pdf

[14] Landa M., Kavka P., Strouhal L., & Cepicky J. (2017). Building a Complete Free and Open Source GIS Infrastructure for Hydrological Computing and Data Publication Using GIS.lab and Gisquick Platforms. International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences, 42. DOI: https://doi.org/10.5194/isprs-archives-XLII-4-W2-101-2017

[15] Vale T., Crnkovic I., De Almeida E. S., Neto P. A. D. M. S., Cavalcanti Y. C., de Lemos Meira S. R., 2016. Twenty-eight years of component-based software engineering. Journal of Systems and Software, 111, 128-148. DOI: https://doi.org/10.1016/j.jss.2015.09.019

[16] JavaScript 3D Library. Available online: https://threejs.org/ (accessed on 20 October 2022)

[17] Martin A., Raponi S., Combe T., Di Pietro R., 2018. Docker ecosystem – Vulnerability Analysis, Computer Communications, Volume 122, Pages 30-43, ISSN 0140-3664, DOI: https://doi.org/10.1016/j.comcom.2018.03.011

APPENDIX A

Tab. A1 - List of objects modeled from the plan documentation.

Id	Village	No.	District	Localization	Exist
1	Bukovina nad Labem	29	Pardubice	50.1220700N, 15.8202439E	no
2	Bukovno	6	Mladá Boleslav	50.4460500N, 14.8392292E	no
3	Francova Lhota	67	Vsetín	49.2079658N, 18.1195658E	no
4	Hlavatce	10	České Budějovice	49.0672028N, 14.2631239E	no
5	Kundratice	60	Semily	50.5815156N, 15.4261467E	yes
6	Náklo	27	Olomouc	49.6525242N, 17.1232369E	no
7	Ostravice	102	Frýdek-Místek	49.5234169N, 18.4025003E	no
8	Petrovice	14	Domažlice	49.4422497N, 12.8915386E	yes
9	Smilkov	14	Benešov	49.6022808N, 14.6168611E	no
10	Šťáhlavice	7	Plzeň-jih	49.6598903N, 13.5249767E	yes
11	Telecí	16	Svitavy	49.6777831N, 16.1960297E	yes
12	Vahlovice	3	Strakonice	49.4460400N, 13.9447031E	yes
13	Zbečno	22	Rakovník	50.0415297N, 13.9207711E	yes
14	Zbožice	2	Havlíčkův Brod	49.6560347N, 15.5633075E	no
15	Zlámanec	12	Uherské Hradiště	49.1284622N, 17.6297361E	no

Tab. A2 - List of objects modeled from physical models.

Id	Village	District	Interier	Note
1	Hlinsko	Chrudim	no	small cottage
2	Orlová	Karviná	yes	wooden cottage
3	Vysoké Mýto	Jablonec nad Nisou	no	town hall