Svistelnyk O. S., student,
Zhytomyr Polytechnic State University

SELECTION AND JUSTIFICATION OF THE ARCHITECTURE FOR THE DEVELOPMENT OF AN ONLINE SERVICE FOR UKRAINIAN LANGUAGE COURSES

In the modern world, the development of online services and applications requires a careful approach to choosing architectural solutions. Regardless of the application, it is important to choose an architecture that optimally meets the needs of users and business. This thesis explores and discusses possible architectural approaches for developing an online service for taking Ukrainian language courses.

The MVC scheme allows you to effectively divide the application components into a model, a view, and a controller, which makes it easier to understand and maintain the code, and speeds up development. The model represents the application data, the view is responsible for displaying information to the user, and the controller processes user requests and interacts with the model and the view. The advantages of this approach are separation of duties, speed of development, and support for simultaneous development by different team members. However, for very simple applications or projects with little functionality, using full MVC may be unnecessary and lead to unnecessary load and increased code complexity. In addition, in some cases, the controller can become a centralized point in the application and overload it from a logical point of view, which can affect the speed and performance of the application.

The MVVM pattern divides the application into a model, a view, and a view-model, where the view-model is responsible for processing data from the model and preparing it for display in the view. This approach uses a declarative approach to programming, which means that the developer describes how the display looks like depending on the data in the view model. The advantages of MVVM are the declarative approach and data binding, which simplifies the development and maintenance of the user interface. However, this approach can lead to an excessive memory load due to the storage of additional view-model objects and create dependence on special frameworks and libraries.

The SPA scheme is an architectural approach in which the entire web application is loaded once, and interaction with the server takes place through asynchronous requests, for example, using AJAX. The entire application code is loaded once during the initial load, and navigation and interaction with the application takes place without a full page reload. This results in a smoother and faster response to user actions and reduces page load times.

The benefits of the SPA approach include an improved user experience, reduced page loads, reduced server load, and more agile development. However, this approach can lead to a higher client-side load and SEO optimization issues for traditional search engines.

Since our goal is to create an online Ukrainian language course service with an improved user experience close to that of a desktop application, we chose to use the SPA (Single-Page Application) approach. Additionally, the use of RESTful API will allow efficient interaction between the server and client parts of the application, transferring data in JSON format, which is a standard for web applications. This approach will simplify the development and expansion of our service, allowing us to quickly respond to changes and add new functionality.

The following main characteristics and principles of RESTful APIs can be distinguished:

- built around resources, which are usually represented as URLs. Each resource has a unique URL, and clients interact with these resources by making HTTP requests to the corresponding URLs;
- RESTful API uses standard HTTP methods to perform various actions on resources.

The general algorithm of the application using the selected architecture is shown in Fig. 1.

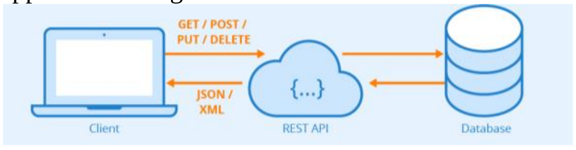

Fig. 1. The general algorithm of the application

After analyzing various architectural approaches, we confirmed that the choice of SPA and RESTful API was optimal for our project and would help us achieve the desired results. This architecture will allow us to create an interactive and convenient service for users, ensure efficient communication between the client and server parts of the application, and quickly adapt to the changing needs of users and the market.

References

1. A. Panchenko, Y. V. Shcherbatyuk. System Analysis and Design of Online Education Systems. Advanced Journal of Education, Science and Technology. 2020. Vol. 2, No. 4. P. 187-194.