

ВИБІР АРХІТЕКТУРИ ПРОГРАМНОЇ КОМПОНЕНТИ ПЛАТФОРМИ ДЛЯ МОНЕТИЗАЦІЇ НАВЧАЛЬНИХ КУРСІВ

¹Вінницький національний технічний університет

Анотація

В даній роботі обґрунтовано вибір та представлено опис архітектури програмної компоненти платформи для монетизації навчальних курсів. Наведено принципи побудови розподіленого мережевого застосунку, використовуючи фреймворк ASP.NET MVC.

Ключові слова: програмна архітектура, сервер, клієнт, запит, ASP.NET, MVC.

Abstract

This work describes the choice of the architecture of the software component of the platform for the monetization of training courses. There is presented the principles of building a distributed network application using the ASP.NET MVC framework.

Keywords: software architecture, server, client, query, ASP.NET, MVC.

Вступ

Одним із найважливіших етапів розробки будь-якого програмного забезпечення є вибір архітектури його програмної компоненти. Багатокористувацький режим платформи для монетизації навчальних курсів передбачає, що архітектура її програмної компоненти повинна забезпечувати велику кількість користувачів, що одночасно хочуть отримати доступ до ресурсів розміщених на цій платформі. Такий вид платформи правильно називати розподіленим мережевим застосунком. Тому адекватний вибір архітектури програмної компоненти для такої платформи є досить актуальною задачею.

Основна частина

Найбільш поширеною концепцією при побудові розподілених мережевих застосунків виділяють архітектуру сервер-клієнт, де сервер виконує усю логіку програми, що пов'язана з взаємодією з базою даних, обробкою запитів, генеруванням відповідей на запити тощо, а клієнт – це оболонка, що доступна кінцевому користувачеві для взаємодії з сайтом [1]. Схему такої архітектури можна побачити на рисунку 1.

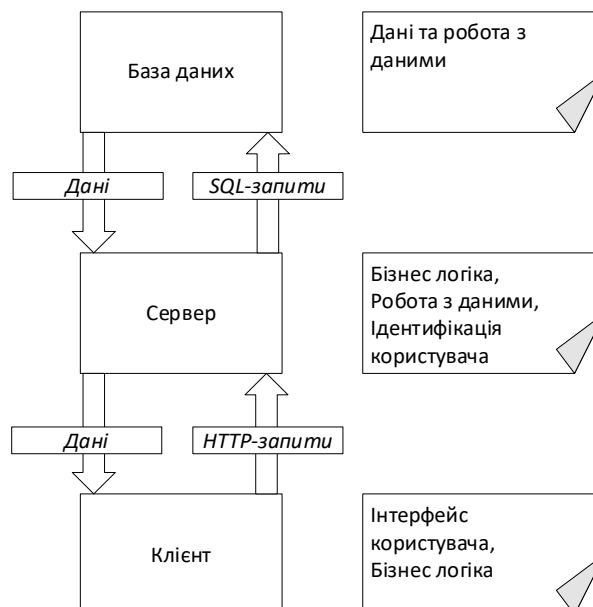


Рисунок 1 – Клієнт-серверна архітектура веб-додатку

Розглянемо компоненти представленої схеми більш детально. Основною програмною частиною веб-додатку є сервер. Він включає в себе всю складну бізнес-логіку та логіку взаємодії з клієнтом та базою даних. Сервер містить механізм ідентифікації користувача. Такий механізм дозволяє забезпечити індивідуальний доступ клієнта до сервера через спеціальне API. Взаємодія з базою даних представленого веб-додатку доступна лише зі сторони сервера, що унеможлиблює небажаний доступ до збережених даних зі сторони клієнта.

Популярним підходом до розробки архітектури веб-додатку є його базування на шаблоні Model-View-Controller. Це шаблон у розробці програмного забезпечення, що зазвичай використовується для реалізації інтерфейсів користувача, даних і логіки керування. Він виділяє поділ між бізнес-логікою програмного забезпечення та її представленням [2].

Найпопулярнішу реалізацію даного підходу представив Microsoft у своєму фреймворку ASP.NET. Розглянемо особливості даної реалізації.

MVC ділить програму на три компоненти – модель, представлення та контролер:

- модель є нічим іншим, як формою даних. У C# аналогом моделі служить клас. Об'єкти моделі зберігають дані, отримані з бази даних;

- представлення – це інтерфейс користувача з допомогою якого можна переглядати дані моделі та змінювати їх. Представлення в ASP.NET MVC – це HTML, CSS і деякий спеціальний синтаксис (синтаксис Razor), який полегшує взаємодію з моделлю та контролером;

- контролер – це клас, що обробляє запити користувача. Такі запити можуть бути ініційовані користувачем з рівня представлення. Контролер обробивши запит повертає нове представлення користувачу як відповідь [3].

Рисунок 2 наочно ілюструє взаємодію між моделлю, представленням і контролером.

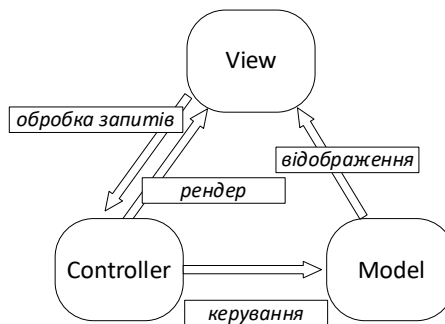


Рисунок 2 – Архітектура MVC веб-додатку

Перебіг запиту користувача в ASP.NET MVC зображено на рисунку 3.

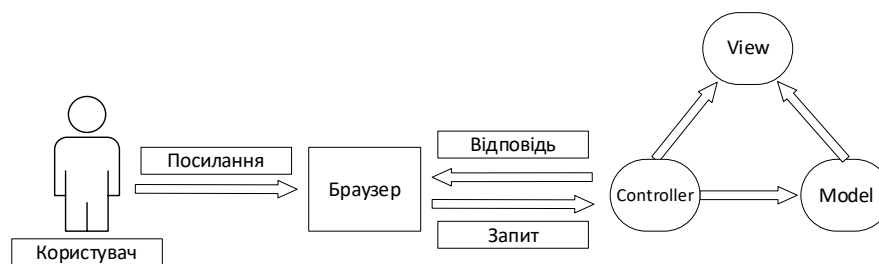


Рисунок 3 – Потік запитів в архітектурі MVC

Після вводу користувачем URL-адреси в пошукову стрічку браузера, вона передається на веб-сервер, де направляється на контролер. Контролер взаємодіє з пов'язаними представленнями та моделями для цього запиту, створює відповідь і надсилає її назад до браузера [3].

Висновок

Для розробки онлайн платформи для монетизації навчальних курсів було обрано MVC архітектуру програмної компоненти, що реалізується через фреймворк ASP.NET. Вона найкраще підходить для вирішення поставлених завдань, дозволяючи створити необхідний для платформи функціонал.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Berson A. Client/server architecture / Alex Berson. – New York: McGraw-Hill, 1992. – 452 с.
2. ASP.NET MVC. [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/aspnet/mvc/>.
3. Freeman A. Pro ASP.NET MVC5 Platform / Adam Freeman., 2014. – 411 с.

Веренько Артем Ігорович – студент групи ЗПІ-18б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: artem.verenco@gmail.com

Романюк Оксана Володимирівна – к.т.н., доцент кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: romaniukoksanav@gmail.com

Artem Verenko – student of group ЗПІ-18b, Faculty for Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: artem.verenco@gmail.com

Oksana Romaniuk – Candidate of Technical Sciences, Associate Professor of the Software Chair, Vinnytsia National Technical University, Vinnytsia, e-mail: romaniukoksanav@gmail.com