

ДОСЛІДЖЕННЯ АРХІТЕКТУРИ ВЕБ-РІШЕНЬ ДЛЯ РЕАЛІЗАЦІЇ СЕРВІСНИХ КОНТРАКТІВ АВТОМОБІЛЬНОЇ КОМПАНІЇ

Алексієв Володимир Олегович, д.т.н., професор

Цемма Дмитро Олександрович, студент магістратури

Харківський національний економічний університет імені Семена Кузнеця, Україна

ВСТУП

Стрімкий розвиток Інтернет-технологій сприяє автоматизації та комп'ютеризації у роботі великих та малих підприємств до яких і відносяться автомобільні компанії дилерські фірми та ін. Поруч із цим стрімко змінюються відповідні архітектурні рішення та технології розробки програмних застосунків, з'являються нові парадигми програмування, які витісняють існуючі. Дослідження спрямоване на визначення оптимальних рішень щодо швидкої розробки веб-застосунків корпоративного рівня, що виконуються у якості внутрішніх сервісів компаній для забезпечення ефективності бізнес-процесів.

Мета роботи – розробка веб-орієнтованого програмного забезпечення для реалізації сервісних контрактів автомобільної компанії. Об'єкт дослідження – процеси формування сервісних контрактів автомобільної компанії. Предмет дослідження – архітектура сучасних веб-рішень.

РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ

Архітектурні рішення – це фундамент для побудови будь-яких веб-додатків. Вибір архітектури веб-рішення фактично визначає ефективність його розробки, розгортання та підтримки протягом життєвого циклу, включно зі здатністю задовольняти надвеликі навантаження на ресурс. Тому, рішенню завдання визначення архітектурних рішень слід приділяти достатньо часу протягом виконання проектних робіт, ставити його на перші місця при розробці [1, 2]. У разі визначення архітектури веб-додатку, першочерговим є безпосередньо врахування бізнес-логіки завдань, що вирішує компанія чи підприємство.

Сервісний контракт є договором щодо надання сервісних послуг на визначений обсяг часу із врахуванням умов експлуатації автомобіля. Це договір між власником та виробником автотранспортного засобу (АТЗ), який дозволяє розширити стандартні гарантійні умови. Наприклад, «Renault Assistance +» дозволяє отримати розширену допомогу в дорозі для автомобілів поза гарантією [3]. Відповідні механізми формування умов розширеної гарантії, визначення її маркетингових переваг та заходів, що пов'язані з забезпеченням виконання договору, визначає складність завдання створення веб-рішення щодо просування сервісу від маркетингової ідеї до сервісного контракту із власником АТЗ. Тому, завдання швидкого створення веб-застосунків корпоративного рівня для роботи у якості внутрішніх сервісів компанії є актуальними та потребують обґрунтування.

Аналізуючи існуючі засоби для розробки веб-рішень та їх популярність у світі, для створення веб-додатків корпоративного рівня можна не брати до уваги мови програмування, які тільки з'являються та ще не пройшли перевірку часом. З іншого боку, застосування рішень за технологією .Net чи Java потребує чіткого та визначеного формування технологічного процесу розробки, що є доцільним для створення надійних та продуктивних систем. У разі створення прототипів веб-сервісів та швидкої розробки веб-рішення, застосування платформи Node.js хоча і є універсальним, однак, у порівнянні із залученням стеку технологій на мові PHP, має деякі обмеження щодо вибору фреймворків та різноманіття архітектурних рішень [4, 5].

Слід виділити PHP-рішення, які мають інфраструктуру на базі чималої кількості фреймворків та не прив'язані до певного рішення операційної системи чи веб-серверу [6]. Зараз розробка, що не ґрунтується на застосуванні фреймворку не є оптимальною та ефективною щодо якості кінцевого продукту. Застосування фреймворку дозволяє скористатися надбанням команд досвідчених розробників та спільноти фахівців, що

розвивають продукт та отримати майже готове рішення на рівні базової структури та архітектури веб-додатку.

Можна виконати аналіз існуючих фреймворків на мові PHP. Серед них слід виділити мікрофреймворки Slim (<https://www.slimframework.com/>) і Silex (<https://silex.symfony.com/>), які доцільно застосовувати для порівняно невеликих проектів, та рішення Yii (<https://www.yiiframework.com/>), Symfony (<https://symfony.com/>) і Laravel (<https://laravel.com/>), що задовольняють вимогам до програмних рішень корпоративного рівня. Для рішення обраної задачі можна обрати Symfony (рис.1), який є одним із найпопулярніших PHP-фреймворків та має архітектуру, яка фактично є збіркою найкращих рішень щодо розробки складних веб-орієнтованих систем [7]. Компоненти фреймворку підключаються до проекту за допомогою сучасного менеджера залежностей Composer (<https://getcomposer.org/>).

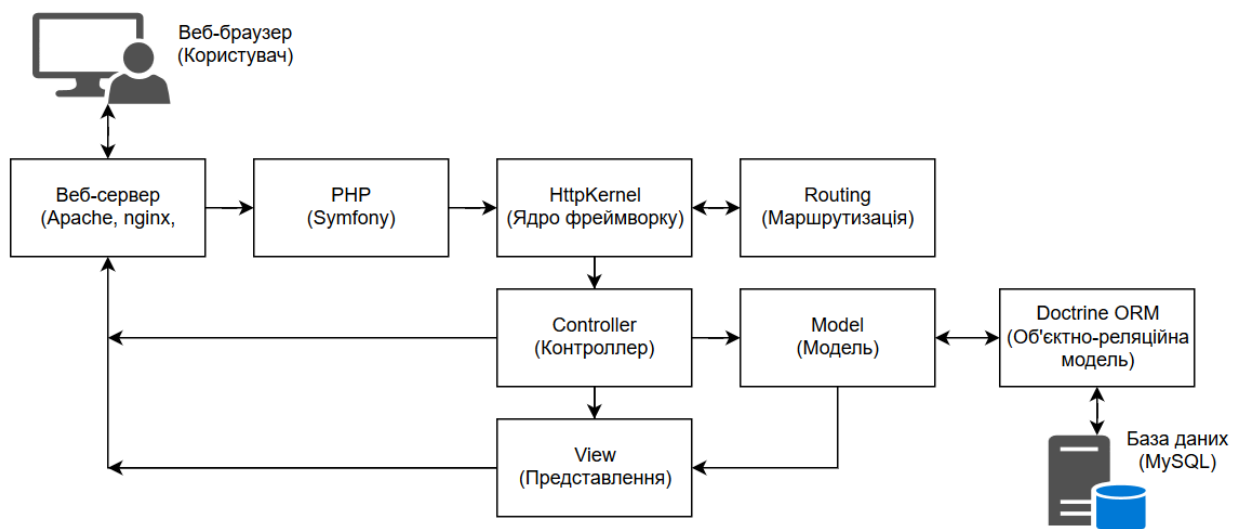


Рисунок 1 – Спрощена структура PHP-фреймворку Symfony

Для роботи з базою даних доцільно обрати СКБД MySQL, як найпопулярніше рішення реляційної БД у зв'язці з мовою PHP, що має порівняно невеликі вимоги до продуктивності середовища виконання (віртуальної машини). Застосування реляційної бази даних обумовлено наявністю чітких зав'язків між даними щодо організації бізнес-процесів компанії, порівняно із невизначеним характером даних, що є характерним у застосуваннях нереляційних систем (NoSQL-рішення), які у більшості використовуються для рішення завдань обробки великих даних (Big Data).

Наступним етапом у визначенні архітектури проектних рішень щодо реалізації сервісних контрактів автомобільної компанії є вибір платформи для виконання завдань: розробки і тестування, розгортання та експлуатації програмного забезпечення. Зараз ефективним та зручним рішенням щодо залучення у якості програмно-апаратної платформи для побудови систем на базі PHP-фреймворку є застосування технології віртуалізації рівня операційної системи Docker [8].

Фактично рішення Docker є програмною платформою для автоматизації розгортання і управління додатками в середовищі віртуалізації. Ця технологія дозволяє «упакувати» додаток з усім його оточенням і залежностями у контейнер, який може бути перенесений на будь-яку Linux-систему з підтримкою cgroups у ядрі, систему Windows із засобами Docker (<https://www.docker.com/>), а також вона надає зручне середовище з управління контейнерів. Таким чином, завдяки використанню Docker можна легко налаштувати автоматичне розгортання (Deploy) веб-додатка, що зараз є не просто популярним рішенням, але й дійсно спрощує виконання адміністративних завдань на стороні сервера щодо забезпечення неперервної роботи веб-рішення.

Також, застосування технології Docker дозволяє отримати майже повний аналог серверу, який виконує обробку даних у реальному часі та в умовах виробничих процесів, безпосередньо на стороні комп'ютеру розробників чи залучити відповідну конфігурацію налаштування для рішення завдань тестування. У разі збільшення навантаження на сервер, який обслуговує рішення з реалізації сервісних контрактів, за умови застосування Docker-контейнерів, є зручним та ефективним «горизонтальне» масштабування веб-рішення завдяки додаванню нових віртуальних вузлів (машин). Таке масштабування можливе у середовищі хмарних сервісів (приватного, публічного чи гібридного), наявності кластерного рішення та засобів Kubernetes (<https://kubernetes.io/>) для керування відповідними процесами.

Слід відзначити, що важливе місце у проекті з розробки веб-рішень займає вибір їх архітектури з точки зору компоновки модулів: монолітна чи мікросервісна [9]. Для рішень із визначеною бізнес-моделлю, раціональним є застосування мікросервісної архітектури та побудови рішення, яке буде складатися із певних сервісів, що будуть взаємодіяти за інтерфейсом REST [10] чи GraphQL (<https://graphql.org/>). У свою чергу, технологія Docker надає ефективні засоби для розгортання та підтримки відповідної інфраструктури.

ВИСНОВКИ

У роботі виконано аналіз проектних рішень щодо створення веб-додатку для реалізації сервісних контрактів автомобільної компанії. Визначена архітектура програмних компонентів за напрямом застосування мікросервісів із фізичною реалізацією на базі Docker-контейнеризації. Запропоновано застосування універсального багатofункціонального PHP-фреймворку Symfony для реалізації програмних рішень та залучення СКБД MySQL у якості зручного та узгодженого сховища даних. Відповідно до зазначених архітектурних рішень виконано розробку прототипу.

ЛІТЕРАТУРА

1. Шкляр Л. Архитектура веб-приложений / Шкляр Леон, Розен Рич; пер. с англ. М.А. Райтмана. – М.: ЭКСМО, 2011. – 640с.
2. Эспозито Д. Разработка современных веб-приложений: анализ предметных областей и технологий / Д. Эспозито; пер. с англ. – М.: Вильямс, 2017. – 464 с.
3. Гарантія та Renault Assistance [Електронний ресурс] / Renault в Україні. – Режим доступу до ресурсу: <https://www.renault.ua/services/renault-service/warranty-insurance-and-assistance.html>.
4. Алексієв В.О. Інформаційно-комунікаційна технологія розроблення транспортно-інформаційного порталу / В.О. Алексієв., В.С. Наумов, М.А. Суховаров, Г.О. Васютіна // Інформаційні технології і мехатроніка. Освіта, наука та працевлаштування: зб. наук. пр.- Х.: Стиль-Издат, 2016. – С. 9–16.
5. Алексієв В. О. Інформаційний розвиток порталу віртуального управління процесами транспортного обслуговування / В. О. Алексієв, О. П. Алексієв // Інформаційні технології: проблеми та перспективи: монографія: / за заг. ред. В. С. Пономаренка. – Х.: Вид-во: Рожко С. Г., 2017. – Розд. 2. – С. 32 – 47.
6. Веллинг Л. Разработка веб-приложений с помощью PHP и MySQL / Л. Веллинг, Л. Томсон; пер. с англ. – М.: Вильямс, 2016. – 768 с.
7. Symfony – Architecture [Electronic resource] / Tutorials Point. – Access mode: https://www.tutorialspoint.com/symfony/symfony_architecture.htm.
8. Моуэт Э. Использование Docker / пер. с англ. А. В. Снастина; науч. ред. А.А. Маркелов. – М.: ДМК Пресс, 2017. – 354 с.
9. Ньюмен С. Создание микросервисов / С. Ньюмен; пер. с англ. – СПб.: Питер, 2016. – 304 с.
10. Web-сервисы RESTful: основы [Электронный ресурс] / IBM developerWorks. – Режим доступа: <https://www.ibm.com/developerworks/ru/library/ws-restfu/index.html>.