

Перевагою даного критерію є його лінійність - він буде збільшуватись пропорційно при зміні глибини нейронної мережі. Крім цього, цей критерій є інтерпретованим, оскільки він зрозуміло пов'язаний з величиною відхилень, які впливають на вихід моделі. З недоліків, варто зазначити нечутливість до “довгих хвостів” розподілів ваг та активацій, через що неможливо в загальному випадку вказати на причину значної похибки. Більш детальний аналіз таких випадків буде проведений в наступних дослідженнях.

ДЖЕРЕЛА

1. Jacob, Benoit, et al. "Quantization and training of neural networks for efficient integer-arithmetic-only inference." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.
2. Nagel, Markus, et al. "A white paper on neural network quantization." *arXiv preprint arXiv:2106.08295* (2021).
3. Lee, Jun Haeng, et al. "Quantization for rapid deployment of deep neural networks." *arXiv preprint arXiv:1810.05488* (2018).
4. Nagel, Markus, et al. "Data-free quantization through weight equalization and bias correction." *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019.

ІНСТРУМЕНТАЛЬНІ ЗАСОБИ РОЗРОБКИ ПРОГРАМНИХ СИСТЕМИ УПРАВЛІННЯ ПЕРСОНАЛОМ / INSTRUMENTAL TOOLS FOR THE DEVELOPMENT OF PERSONNEL MANAGEMENT SOFTWARE SYSTEMS

Ткаченко В.А. / Tkachenko V.A.

Київський національний університет імені Тараса Шевченка / Taras Shevchenko National University of Kyiv

01601, Київ, вул. Володимирська, 64/13, E-mail: tkachenkovitaliy@knu.ua

The article discusses software development tools for personnel management systems, with an emphasis on ensuring data security and operational efficiency. Particular attention is paid to the use of qualified electronic signatures (QES) to identify employees and guarantee the legality of operations, especially in the context of increased cybersecurity threats. The role of the Public Key Infrastructure (PKI) in ensuring information security and authentication is described. The article highlights the importance of choosing a platform for application development, especially .NET and ASP.NET Core, to create secure, high-performance web applications. The article also discusses the use of the Blazor framework for developing cross-platform solutions, including Blazor Server, Blazor WebAssembly, and Blazor Web App, which provide various rendering options and development flexibility. All these tools allow you to create reliable, scalable workforce management systems that meet security and efficiency requirements.

Системи управління персоналом є критично важливими для оптимізації процесів управління людськими ресурсами, забезпечення дотримання трудових норм і підвищення ефективності організації. У цифрову еру розробка таких систем вимагає надійних інструментів, які підтримують різноманітні функціональні можливості, включаючи керування даними, безпеку та інтеграцію із зовнішніми службами.

Одним з ключових аспектів розробки системи управління персоналом є забезпечення безпеки даних, особливо в конфіденційних областях, таких як особиста інформація співробітників, і оптимізацію зручності використання як для адміністраторів, так і для кінцевих користувачів. В умовах підвищених загроз дотримання національних стандартів цифрових підписів, таких як кваліфіковані електронні підписи (КЕП), додає додатковий рівень безпеки персональних даних. КЕП це удосконалений електронний підпис, який створюється з використанням засобу кваліфікованого електронного підпису і базується на кваліфікованому сертифікаті відкритого ключа. Він містить інформацію про власника (працівника), а також криптографічні ключі та дані сертифікаційного центру, який їх видає. Також КЕП має таку саму юридичну силу, як і власноручний підпис.

КЕП дозволяє ідентифікувати працівника під час аутентифікації, базуючись на даних КЕП система визначає роль працівника, наприклад бухгалтер, кадровик, а також надає відповідний доступ до даних та функціоналу який визначений у рольовій моделі системи. Також КЕП гарантує дотримання законодавства про захист інформації та електронні довірчі послуги.

У системах управління персоналом КЕП використовується для забезпечення захищеного входу до інформаційної системи, підписання електронних документів, наприклад наказів та заяв.

Рольова модель доступу, яка визначається через КЕП, надає можливість забезпечити виконання лише тих функцій які передбачені посадовими обов'язками працівника що мінімізує ризики несанкціонованого доступу до конфіденційних даних.

В Україні для роботи з КЕП використовується Інфраструктура відкритих ключів (Public Key Infrastructure (PKI)). PKI — це система управління ключами, яка містить програмне забезпечення, сертифікаційні центри та криптографічні алгоритми для забезпечення аутентифікації та захисту інформації. PKI працює за принципом пар ключів, публічного та приватного.

Приватний ключ використовується для підписання документів або виконання операцій.

Приватний ключ ніколи не передається іншим особам.

Публічний ключ дозволяє перевірити підписані дані або зашифровану інформацію, використовуючи алгоритми криптографії.

Ще один критичний аспект розробки системи управління персоналом є вибір платформи для розробки застосунків. Платформа .NET надає можливість розробляти безпечні, кросплатформні, високо навантажені застосунки, а також має велику екосистему.

Одним з компонентів цієї екосистеми є фреймворк ASP.NET Core який використовується для розробки кросплатформних веб-застосунків. ASP.NET Core це кросплатформна, високопродуктивна структура з відкритим вихідним кодом для створення сучасних хмарних застосунків, підключених до Інтернету.

Для розробки веб-застосунків у структурі ASP.NET Core використовується Blazor. Blazor — це веб-фреймворк, який використовує мову програмування C# а також має можливість рендерінгу як на стороні сервера, так і на стороні клієнта. Оскільки Blazor є частиною ASP.NET Core він має можливість використовувати існуючі компоненти екосистеми .NET з бібліотеками .NET. Blazor має три шаблони, Blazor Server, Blazor WebAssembly, Blazor Web App. Кожен з яких надає можливість рендерити компоненти по-різному.

Blazor WebAssembly повністю працює в браузері та використовує WebAssembly (Wasm) для виконання коду .NET. Blazor WebAssembly може використовувати два варіанти компіляції. Ahead of Time (AOT) надає можливість скомпілювати код .NET безпосередньо в Wasm. Без використання AOT Blazor WebAssembly можна запускати в браузері за допомогою інтерпретатора мови .NET Intermediate Language, реалізованого в Wasm з частковою підтримкою компіляції just-in-time (JIT).

Застосунки на платформі Blazor Server розміщують компоненти Blazor на сервері та обробляють взаємодії призначеного для користувача інтерфейсу за допомогою з'єднання SignalR в реальному часі. Коли користувач взаємодіє з застосунком, події призначеного для користувацького інтерфейсу відправляються на сервер за допомогою з'єднання. Коли компонент обробляє подію призначену для користувацького інтерфейсу, вона відображається на основі свого оновленого стану. Blazor порівнює недавно відрендерений висновок з раніше відрендереним, відправляє зміни назад в браузер і застосовує їх до DOM.

Blazor Web App поєднує два варіанти рендерінгу та надає можливість використовувати кожен з варіантів рендерінгу до окремої сторінки або компоненту, що надає гнучкість при розробці веб-застосунка.

Можливість Blazor Server обробляти дані на сервері, використовувати переваги платформи

.NET такі як бібліотеки та мову програмування C#, робить Blazor Server перспективним рішенням для використання у якості фреймворку для розробки систем управління персоналом та інших застосунків.

Одним з ключових аспектів розробки якісного програмного забезпечення є підбір засобів розробки, що відповідатимуть вимогам проекту, зможуть забезпечити ефективність, зручність у виконанні, а також забезпечать необхідний рівень безпеки, надійність та масштабованості продукту, дозволять адаптуватися до вимог бізнесу.

СЕКЦІЯ 4. ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

РОЗПОДІЛ КВОТ ХМАРНИХ РЕСУРСІВ ЯК ПРИКЛАД КООПЕРАТИВНОЇ ГРИ / CLOUD QUOTES ON RESOURCES ALLOCATION AS A CORPORATE GAME

Артюшенко Б.А., к.т.н. / Bohdan Artiushenko, PhD

консультант ТОВ «Інститут інформаційних технологій «Інтеліас»» / consultant at
“Institute of Information Technologies “Intelias LLC””
79032, м. Львів, вулиця Шафарика, 16/74 / 79032, 16/74 Schafaryka St., Lviv, Ukraine
E-mail: bogdan.artiushenko@intellias.com

This work explores the modern approach to cloud resource allocation, focusing on how computational resources are distributed between different subprojects or product teams. The process is modeled as a game, where teams with competing priorities must negotiate based on their claims and the available budget. The ultimate objective is to optimize resources allocation for the overall benefit of the organization, without unnecessary undermining individual team needs. A penalty method approach is used to model the allocation of computational resources effectively. The problem considers the constraints of resource availability, team priorities, and the need to prevent conflicts or blockages within a system. The proposed approach involves optimization techniques that ensure efficient allocation while maintaining project integrity and overall organizational goals.

Хмарні технології дозволяють надавати обчислювальні потужності як послугу. З їх допомогою організація-користувач може більш гнучко розподіляти обчислювальні потужності, масштабувати та знизити початкові інвестиції в обчислювальні системи. Разом з цим їх використання може бути пов'язане з неконтрольованим зростанням фінансових витрат та вимог до обчислювальних ресурсів. [1] Якщо над проектом-системою працює кілька команд розробників то це може призвести до блокування всього проекту однією з команд, яка використовує всі наявні ресурси. Статись це може і без відома цієї команди в наслідок автоматичного виділення ресурсів хмарою. Одним із підходів для запобігання подібних блокувань є впровадження квот на команди. Наприклад, система управління застосунками в контейнерах Kubernetes дозволяє виставляти по-командні квоти для оперативної та постійної пам'яті, центрального процесора, викликів API системи [2]. Кожна команда має власні пріоритети, відповідає за свої частини проекту, мікросервіси, контейнери. І, відповідно, зацікавлена переважно у власних результатах а не системи в цілому. Команда може запитати потрібні ресурси у особи що виділяє квоти. Вважатимемо що команди використовують однаковий підхід для оцінки потреб в ресурсах.