

Технічні науки

УДК 004.02 + 616.1

**Гриневич Дмитро Ігорович**

*студент*

*Національного технічного університету України*

*«Київський політехнічний інститут імені Ігоря Сікорського»*

**Гриневич Дмитрий Игоревич**

*студент*

*Национального технического университета Украины*

*«Киевский политехнический институт имени Игоря Сикорского»*

**Hrynevych Dmytro**

*Student of the*

*National Technical University of Ukraine*

*«Igor Sikorsky Kyiv Polytechnic Institute»*

**Науковий керівник:**

**Яковенко Альона Вікторівна**

*кандидат технічних наук,*

*доцент кафедри біомедичної інженерії*

*Національний технічний університет України*

*"Київський політехнічний інститут імені Ігоря Сікорського"*

**АНАЛІЗ ПІДХОДІВ ДО СТВОРЕННЯ СИСТЕМИ АВТОМАТИЗАЦІЇ  
КОНТРОЛЮ ТА УПРАВЛІННЯ ХОДОМ ВИКОНАННЯ  
ДИПЛОМНИХ РОБІТ**

**АНАЛИЗ ПОДХОДОВ К СОЗДАНИЮ СИСТЕМЫ  
АВТОМАТИЗАЦИИ КОНТРОЛЯ И УПРАВЛЕНИЯ ПРОЦЕССОМ  
ВЫПОЛНЕНИЯ ДИПЛОМНЫХ РАБОТ**

## ANALYSIS OF APPROACHES FOR AUTOMATION CONTROL AND DIPLOMA GRADUATION PROCESS MANAGEMENT

**Анотація.** В статті розглянута розробка системи контролю, обліку та автоматизації подавання дипломної роботи до захисту з використанням платформи .NET, мови програмування F#, фреймворку для побудови веб-додатків ASP.NET Core MVC та Elasticsearch. Описана загальна архітектура додатку та деякі її більш конкретні фрагменти, що варті уваги. Проаналізовані плюси і мінуси обраних технологій.

**Ключові слова:** автоматизація, документообіг, .NET, ASP.NET Core, Elasticsearch, ASP.NET Core MVC, F#.

**Аннотация.** В статье рассмотрена разработка системы контроля, учета и автоматизации подготовки дипломной работы к защите с использованием платформы .NET, языка программирования F#, фреймворка для построения веб-приложений ASP.NET Core MVC и Elasticsearch. Описана общая архитектура приложения и некоторые ее более конкретные фрагменты, которые стоят внимания. Проанализированы плюсы и минусы выбранных технологий.

**Ключевые слова:** автоматизация, документооборот, .NET, ASP.NET Core, Elasticsearch, ASP.NET Core MVC, F#.

**Summary.** The article describes development of control, accounting and process of diploma graduation system with using of .NET platform, F# programming language, ASP.NET Core MVC framework for web applications and Elasticsearch. It describes whole system architecture with diving into its important fragments. Good and bad sides of the chosen technologies are analyzed.

**Key words:** automation, document flow, .NET, ASP.NET Core, Elasticsearch, ASP.NET Core MVC, F#.

**Постановка проблеми.** Процес інформатизації вищої освіти супроводжується активним впровадженням засобів інформаційних та комунікаційних технологій в різні предметні області, в професійну діяльність викладачів та організацію управління навчально-виховним процесом. Одним з таких процесів є написання та подання дипломної роботи до захисту в українських ВУЗах.

Учасники, які задіяні під час роботи над дипломною роботою чи проектом працюють над великою кількістю різних документів та самих учасників під час роботи над однією дипломною роботою досить велика кількість, при чому іноді виникає проблема версійності документів та порушується правильний порядок проходження всіх етапів щодо подання дипломної роботи до захисту.

**Аналіз останніх досліджень і публікацій.** Програмування на F# описував Isaac Abraham [1], побудову архітектури додатків на функціональній мові програмування розглядав Scott Wlaschin [4]. Правильне використання та конфігурування Elasticsearch описали у своїй книзі Clinton Gormley [3] та Zachary Tong [3]. Розробку за допомогою ASP.NET Core MVC розглянув Adam Freeman [2]. Використовування докер було описано Еріаном Моуетом [5].

**Мета дослідження:** створення автоматизованої системи контролю та управління ходом виконання дипломних робіт, що дозволяє науковим керівникам, нормоконтролеру, секретарю державної екзаменаційної комісії (ДЕКу) та іншим взаємодіяти зі студентами протягом підготовки дипломних робіт.

**Виклад основного матеріалу.** Система, що була розроблена автоматизує процеси документообігу: ведення дипломних робіт студентів, їх наукових публікацій, актів впровадження, гарантійних листів тощо та організацію інформаційної взаємодії між користувачами системи.

У разі типу додатку була обрана веб-версія, оскільки це є найбільш зручним варіантом з боку використання систем такого типу.

Програма представляє собою ASP.NET Core MVC додаток, який має трирівневу архітектуру: шар представлення, шар предметної логіки та шар доступу до даних. У разі мови програмування була обрана F#. У якості сховища для даних використовувався Elasticsearch. Шар представлення написаний за допомогою ASP.NET Core MVC, шар предметної логіки є .NET Standart бібліотекою класів на F#, шар доступу до даних використовує клієнт Elasticsearch для .NET NEST. Реалізовано також сервіс нотифікацій, що дозволяє відправляти електронні листи іншим учасникам дипломного проектування у ролі background worker та модуль аутентифікації.

Elasticsearch може бути розгорнутим з Docker контейнера або як сервіс у хмарі Amazon Web Services.

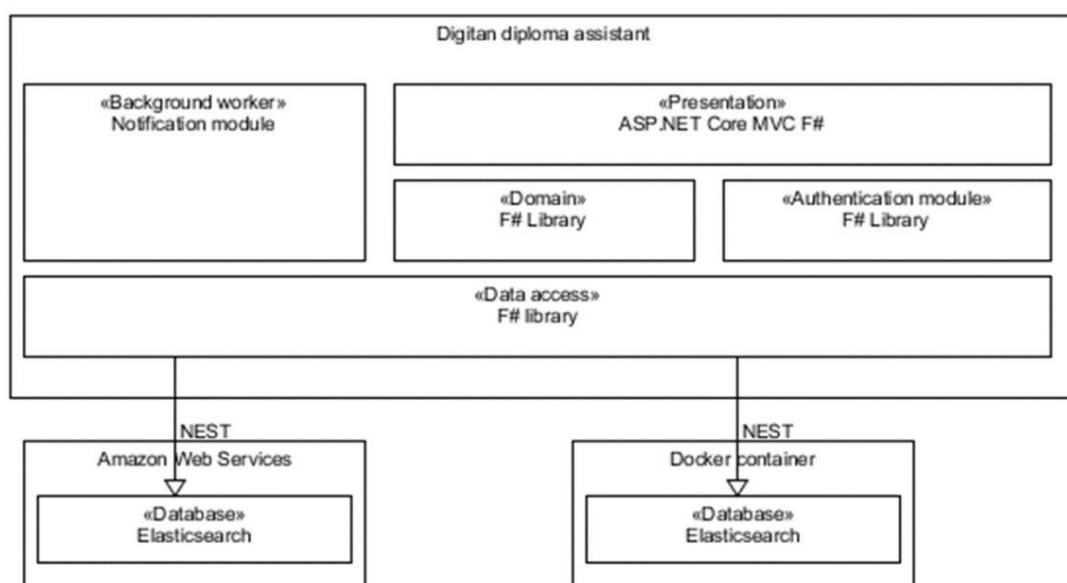


Рис. 1. Архітектура системи

При побудові шару представлення було знайдено сильні та слабкі сторони ASP.NET Core MVC у парі з F#. До сильних сторін можна віднести стабільну роботу ASP.NET Core MVC та F#, наявність великого ASP.NET

Core ком'юніті та підтримкою фреймвоку Microsoft та open-source. До слабких сторін відноситься поєднання об'єктно-орієнтованої парадигми та функціональної, зокрема в controller-частинах MVC фреймворку, інфраструктурної та Razor.

Controller сам по собі є класом, тож поєднання шаблону Partial application з функціонального програмування для впровадження залежностей функцій шару предметної логіки та доступу до бази даних, що базується на основі currying, з впровадженням залежностей, коли controller вимагає для свого існування або конфігурування якісь компоненти або параметри, є невід'ємною частиною ASP.NET Core MVC F#.

Модуль автентифікації містить логіку автентифікації користувачів та використовує окремий індекс у Elasticsearch для збереження необхідних даних користувачів та їх ролі.

Шар доступу до даних використовує NEST-клієнт .NET для Elasticsearch, що написаний на C#, тож було необхідно реалізувати F#-адаптер для C# коду, щоб мати можливість писати код у функціональному стилі.

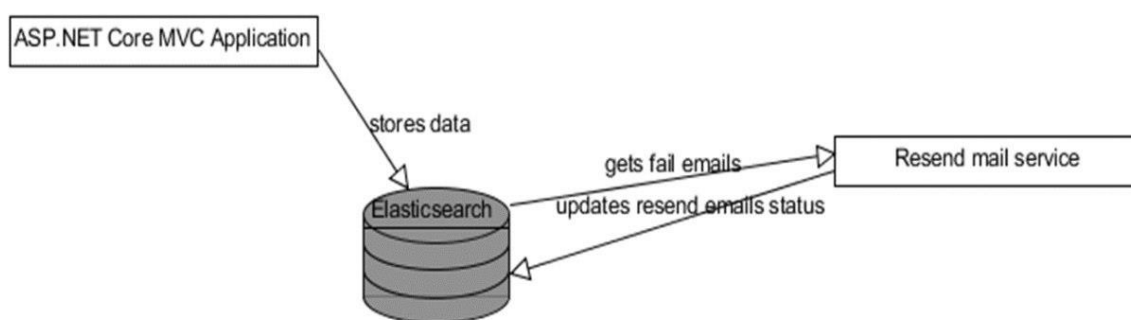


Рис. 2. Принцип роботи сервіса нотифікацій

Сервіс нотифікацій реалізовано окремо від веб додатку, щоб не затримувати повернення запиту користувачеві. Веб-додаток складає в базу

даних необхідні дані для сервісу нотифікацій, а останній, в свою чергу, отримує її з бази та відправляє. У разі помилки відправлення нотифікації буде відіслано ще раз через певний проміжок часу.

Elasticsearch був сконфігурованим таким чином, щоб мати можливість робити успішний повнотекстовий пошук, тож було встановлено плагін для українського аналізатора. Також на кожний логічний тип даних створювався окремий індекс. Для документів, що часто оновлюються, був використаний підхід application side joins, для інших – nested. Це пов'язано з тим, що використовуване документноорієнтоване сховище передбачає певне копіювання даних та оптимізоване для цього, а при зміні будь-якої властивості документу він переіндексується.

**Висновки.** Загальна архітектура системи з автоматизації контролю та управлінням за ходом виконання дипломних робіт зроблена правильно – не викликає додаткових складнощів під час розробки та має чіткі границі шарів та контекстів. Elasticsearch легко розгортається як з Docker контейнеру, так і в хмарі AWS. F# як функціональна мова програмування показала себе добре, надаючи змогу писати легкий та читабельний код, але сама платформа .NET ще не має стільки готових рішень для F# як для C#, тож для деяких речей доводилося реалізовувати F# адаптери та саме використання F# в ASP.NET Core MVC потребувало навичок, як з функціонального, так і з об'єктно-орієнтованого програмування.

### **Література**

1. Abraham I. Get programming with f# / I. Abraham. 2018. 594 p.
2. Freeman A. Pro asp.net core mvc / A. Freeman. 2016. 1031 p.
3. Gormley C. Elasticsearch: the definitive guide a distributed real-time search and analytics engine / C. Gormley, Z. Tong. 2015. 724 p.
4. Wlaschin S. Domain modeling made functional / S. Wlaschin. 2017. 297 p.
5. Моуэт Э. Использование Docker / Э. Моуэт. Москва : 2017. 356 p.

## **References**

1. Abraham I. Get programming with f# / I. Abraham. 2018. 594 p.
2. Freeman A. Pro asp.net core mvc / A. Freeman. 2016. 1031 p.
3. Gormley C. Elasticsearch: the definitive guide a distributed real-time search and analytics engine / C. Gormley, Z. Tong. 2015. 724 p.
4. Wlaschin S. Domain modeling made functional / S. Wlaschin. 2017. 297 p.
5. Moyet E. Ispolzovanie Docker / E. Moyet. Москва: 2017. 356 p.