

UNIWERSYTET ŚLĄSKI W KATOWICACH
WYDZIAŁ NAUK ŚCISŁYCH I TECHNICZNYCH

Kacper Hałaczekiewicz

Nr albumu: 338889

Zastosowanie narzędzi DevOps
na przykładzie aplikacji do nawigacji po budynku

PRACA DYPLOMOWA INŻYNIERSKA

Promotor: dr hab. Arkadiusz Bubak

Katowice 2022

Contents

1	Wstęp	3
2	Motywacja	4
3	Objaśnienie narzędzi	5
3.1	Visual Studio	5
3.2	Język C#	5
3.3	MVVM	5
3.4	.NET MAUI	5
3.5	Microsoft Azure	6
3.6	Fluent Assertions	6
3.7	*Biblioteka pathfinding*	6
3.8	GIT	6
4	Aplikacja SMCEBI Navigator	7
5	Zastosowanie platformy Microsoft Azure	8
6	Dodatkowe źródła i przydatne materiały	9

1 Wstęp

Założeniem pracy jest stworzenie kompletnego rozwiązania w postaci aplikacji ułatwiającej nawigowanie po budynku Śląskiego Międzyuczelnianego Centrum Edukacji i Badań Interdyscyplinarnych (SMCEBI), oraz zautomatyzowanego środowiska kompilującego i dystrybuującego rozwiązanie.

W mojej pracy zawarłem krótkie objaśnienia terminów, których zrozumienie jest bardzo przydatne (a czasem nawet kluczowe) do uruchomienia mojego rozwiązania. Najciekawsze źródła, których nie cytowałem, ale odwiedziłem i naprowadziły mnie na poprawne tory bądź rozwijają temat umieściłem w rozdziale

Podczas tworzenia programu skorzystałem z gotowej biblioteki do wyznaczania ścieżek, wobec czego w niniejszej pracy znajduje się wyłącznie skrócone wytłumaczenie schematu działania jej algorytmu.

2 Motywacja

Zastosowanie narzędzi wspomagających tworzenie, testowanie, wydawanie i opiekę nad produktem znacząco wpływa na całe rozwiązanie. Do ich zalet należy przyspieszenie prac, poprawa jakości finalnego produktu oraz odciążenie pracowników z wykonywania powtarzalnych czynności (które są powodem najczęstszych błędów i niedopatrzeń). Skorzystanie z ich dobrodziejstw skutkuje wyraźnym poprawieniem komfortu pracy, wydajności zespołu i przede wszystkim jakości kodu.

3 Objaśnienie narzędzi

3.1 Visual Studio

Visual Studio [1] to zintegrowane środowisko programistyczne (ang. IDE), czyli połączenie edytora tekstu, kompilatora oraz debuggera, jak również wielu narzędzi przydatnych podczas rozwijania oprogramowania (uproszczony refactoring, analiza zużycia zasobów, rozszerzone debugowanie). Pakiety narzędzi dostępne są dla wielu języków programowania, co czyni z niego jed

W moim projekcie wykorzystałem wersję 2022 - na dzień wydania pracy jest to najnowsza edycja, pozwala na szybką konfigurację środowiska do celów kompilacyjnych.

3.2 Język C#

C# [2] (z ang. wym. "See Sharp", pol. "Si Szarp") to wydajny język obiektowy wysokiego poziomu, stworzony przez Microsoft. Należy do rodziny języków C i jest uniwersalny w zastosowaniu.

3.3 MVVM

MVVM to akronim Model-View-Viewmodel (w tłumaczeniu Model - Widok - Model widoku), czyli programistycznego wzorca projektowego. Polega na wprowadzeniu separacji pomiędzy modelem danych i sposobem ich obróbki, widokami je reprezentującymi, oraz operacjami pomiędzy nimi. Jego główną zaletą jest możliwość oddzielenia ogólnych założeń programu od szczegółów integracji z platformą - w projekcie [3] jest to wyraźnie zarysowane dzięki przyjętej konwencji nazewnictwa plików (postfix -Model, -ViewModel lub -View). Na rysunku 1 możemy zaobserwować zależność komunikacji klas stworzonych z wykorzystaniem właśnie tego wzorca.

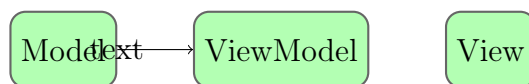


Figure 1: Zależności między klasami w MVVM [4]

3.4 .NET MAUI

.NET MAUI [5] to platforma programistyczna (ang. framework) umożliwiająca kompilację natywnych aplikacji wieloplatformowych. Jest to następca Xamarina [6], czyli poprzedniego rozwiązania Microsoftu do tych samych

zastosowań. Najważniejszymi różnicami jest obsługa projektów - Xamarin wymagał osobnych projektów na każdy system operacyjny, zaś MAUI obsługuje je wewnątrz jednego, uniwersalnego projektu.

Framework umożliwia skorzystanie z kontrolek i interfejsów każdego z systemów za pomocą bezpośrednich odniesień do jego API, a następnie na etapie kompilacji wszelkie odniesienia są przebudowywane na odpowiednie interfejsy każdej platformy.

3.5 Microsoft Azure

Microsoft Azure [7] to internetowa platforma oferująca szeroki wachlarz narzędzi i usług z zakresu tworzenia, rozwoju i dystrybucji oprogramowania. Za jej pomocą możemy wynająć moc obliczeniową w chmurze Microsoftu - począwszy od baz danych, aż po maszyny wirtualne. Platforma oferuje również pełne środowisko programistyczne - DevOps Server, które wspomaga komunikację, formalizujące wprowadzane zmiany, automatycznie testuje, integruje i dystrybuje rozwiązania.

3.6 Fluent Assertions

Fluent Assertions [8] to narzędzie dodające do IDE specjalne rozszerzenia metod, które wykorzystywane są podczas testowania oprogramowania przed jego wydaniem. Jego przewagą nad standardowymi rozwiązaniami jest większy wybór i czytelniejsze nazewnictwo funkcji, jak również wsparcie aktywnej społeczności.

3.7 *Biblioteka pathfinding*

<Tutaj skrótowo opiszę skąd pobrałem bibliotekę, skrótowo opiszę działanie wykorzystywanego algorytmu>

3.8 GIT

Rozproszony system kontroli wersji pozwala na pracę dowolnej ilości osób na dowolnej ilości urządzeń jednocześnie, przy ciągłym utrzymaniu aktualności kodu. GIT z wielu zatwierdzonych części kodu (ang. commitów) podzielonych na gałęzie (ang. branch) tworzy drzewo zmian, po którym programiści mogą swobodnie poruszać się, zaglądać do dawnych wersji, pomimo braku przechowywania każdej z nich na swoim dysku. System może być hostowany na własnym serwerze lub chmurze (przykładowo wykorzystany w tym projekcie GitHub [3, 9]), ale przechowywanie repozytorium lokalnie jest również możliwe.

4 Aplikacja SMCEBI Navigator

% Opiszę wstępne założenia aplikacji, pokażę najistotniejsze elementy implementacji:

- na grafice zwizualizuję wielopoziomowy pathfinding
- przedstawię wygląd interfejsu
- przedstawię sposób działania przejść między widokami (Shell Navigation)
- Pokażę architektoniczny podział aplikacji na API oraz aplikację (jeszcze nie wiem czy aplikacja sama w sobie powinna liczyć ścieżki, czy będą to zapytania do server-hosted aplikacji)
- pokażę strukturę pliku, podkreślę zastosowanie wzorca MVVM

5 Zastosowanie platformy Microsoft Azure

% W tym rozdziale planuję opisać w jaki sposób tworzone są kompletne pipeline'y, dlaczego są one tak istotne (najlepiej na przykładzie anegdotycznym?). Po krótkim wstępie zaprezentuję jak skonfigurowany jest mój pipeline, na jakie elementy należy zwrócić uwagę, jaki jest efekt końcowy jego działania.

6 Dodatkowe źródła i przydatne materiały

% Tutaj zamierzam umieścić przydatne linki, wraz z krótkimi opisami ich zawartości, które przyczyniły się w istotny sposób do poprawy jakości pracy, ale nie wypisałem ich w cytowaniach, ze względu na brak bezpośredniego połączenia z treścią.

Przykładowo:

- YouTube - .NET MAUI for Beginners oficjalny poradnik, który wstępnie oprowadził mnie po zastosowaniach i obsłudze frameworka. Bardzo przydatny dla każdego początkującego użytkownika twórcy w tej technologii.

References

- [1] Microsoft. *Microsoft Visual Studio - Home Page*. URL: <https://visualstudio.microsoft.com/pl/> (visited on 06/09/2023).
- [2] Microsoft. *C# documentation*. URL: <https://learn.microsoft.com/en-us/dotnet/csharp/> (visited on 06/09/2023).
- [3] Kacper Hałaczkiwicz. *SMCEBI Navigator - repozytorium na GitHub*.
- [4] Microsoft. *The MVVM Pattern*. URL: [https://learn.microsoft.com/en-us/previous-versions/msp-n-p/hh848246\(v=pandp.10\)?redirectedfrom=MSDN](https://learn.microsoft.com/en-us/previous-versions/msp-n-p/hh848246(v=pandp.10)?redirectedfrom=MSDN) (visited on 06/09/2023).
- [5] Microsoft. *Microsoft .NET MAUI - Home Page*. URL: <https://learn.microsoft.com/en-us/dotnet/maui/> (visited on 06/09/2023).
- [6] Microsoft. *Microsoft Xamarin - Home Page*. URL: <https://dotnet.microsoft.com/en-us/apps/xamarin> (visited on 06/09/2023).
- [7] Microsoft. *Microsoft Azure - Home Page*. URL: <https://learn.microsoft.com/en-us/azure/?product=devops> (visited on 06/09/2023).
- [8] Dennis Doomen. *Fluent Assertions - Home Page*. URL: <https://fluentassertions.com/> (visited on 06/09/2023).
- [9] GitHub. *GitHub - Version Control hosting*. 2020. URL: <https://github.com/> (visited on 07/05/2022).