



Politechnika Wrocławska

Wydział Informatyki i Zarządzania

kierunek studiów: Informatyka

Praca dyplomowa - inżynierska

Wieloplatformowa aplikacja mobilna do zakupu biletów kinowych w technologii Xamarin

Marceli Grabowski

słowa kluczowe:

1 linia

2 linia

3 linia

krótkie streszczenie:

Praca opisuje projekt i implementację aplikacji mobilnej na platformy Android oraz iOS, umożliwiającej zakup biletów kinowych, przy użyciu najnowszych rozwiązań z platformy .NET.

opiekun pracy dyplomowej	dr inż. Zbigniew Fryźlewicz <i>Tytuł/stopień naukowy/imię i nazwisko</i> <i>ocena</i> <i>podpis</i>
Ostateczna ocena za pracę dyplomową			
Przewodniczący Komisji egzaminu dyplomowego <i>Tytuł/stopień naukowy/imię i nazwisko</i> <i>ocena</i> <i>podpis</i>

Do celów archiwalnych pracę dyplomową zakwalifikowano do:*

a) kategorii A (akta wieczyste)

b) kategorii BE 50 (po 50 latach podlegające ekspertyzie)

* niepotrzebne skreślić

pieczęć wydziałowa

Wrocław 2017

Spis treści

Streszczenie	4
Wstęp	6
1 Wprowadzenie do problematyki	7
2 Technologie i narzędzia wykorzystywane w pracy	8
2.1 Xamarin.Android	8
2.2 Xamarin.iOS	8
2.3 ASP.NET Core	8
2.4 Entity Framework Core	8
2.5 Dapper	8
2.6 MVVMCross	8
2.7 AutoFac	8
2.8 Team Foundation Server	8
2.9 HockeyApp	8
2.10 Microsoft Azure	8
3 Założenia projektowe	9
3.1 Przedmiot pracy	9
3.2 Wymagania funkcjonalne	9
3.3 Wymagania нефункционалне	11
3.4 Opis podstawowej architektury systemu	11
4 Projekt aplikacji	13
4.1 Przypadki użycia	13
4.2 Interfejs	13
4.3 Diagram klas	13

5	Implementacja	14
5.1	DevOps	14
5.2	Autoryzacja użytkowników aplikacji	14
5.3	Synchronizacja danych offline-online	14
5.4	Bezpieczeństwo aplikacji	14
5.5	Implementacja wzorca CQRS	14
5.6	Testy interfejsu aplikacji	14
6	Podsumowanie	15
	Bibliografia	16
	Załączniki	17
	Spis tabel	17
	Spis rysunków	17
	Spis listingów	17
	Instrukcja kompilacji i testowego uruchomienia aplikacji	17

Streszczenie

Streszczenie

Abstract

Abstract

Tu walnij sobie dedykację

Wstep

Rozdział 1

Wprowadzenie do problematyki

Rozdział 2

Technologie i narzędzia wykorzystywane w pracy

- 2.1 Xamarin.Android
- 2.2 Xamarin.iOS
- 2.3 ASP.NET Core
- 2.4 Entity Framework Core
- 2.5 Dapper
- 2.6 MVVMCross
- 2.7 AutoFac
- 2.8 Team Foundation Server
- 2.9 HockeyApp
- 2.10 Microsoft Azure

Rozdział 3

Założenia projektowe

3.1 Przedmiot pracy

Przedmiotem pracy jest utworzenie aplikacji mobilnej na platformy Android oraz iOS, umożliwiającej rezerwację i zakup biletów kinowych w ramach sieci kin, wraz z towarzyszącą aplikacją serwerową. Aplikacja kliencka będzie utworzona w oparciu o platformę Xamarin, natomiast aplikacja serwerowa w oparciu o framework ASP.NET Core

3.2 Wymagania funkcjonalne

System powinien spełniać następujące wymagania funkcjonalne. Przy definiowaniu wymagań przyjęto następujących aktorów - Klient, Pracownik kina, Administrator systemu, System

1. Klient ma możliwość stworzenia konta użytkownika na podstawie adresu e-mail, w celu zachowania preferencji użytkownika pomiędzy urządzeniami.
2. Klient ma możliwość stworzenia konta użytkownika z wykorzystaniem konta w jednym ze wspieranych portali społecznościowych (Facebook, Twitter, Google), w celu przyspieszenia procesu tworzenia konta.
3. Klient ma możliwość modyfikacji informacji o koncie użytkownika.
4. Klient ma możliwość zresetowania hasła do konta użytkownika, w celu odzyskania dostępu do konta.
5. Klient ma możliwość wyboru domyślnego kina, w celu łatwiejszego dostępu do aktualnego repertuaru.
6. Klient ma możliwość przeglądania aktualnego repertuaru w danym kinie.
7. Klient ma możliwość przeglądania podstawowych informacji o filmie z repertuaru.
8. Klient ma możliwość złożenia rezerwacji biletu(ów) na wybrany seans w wybranym kinie.
9. Klient ma możliwość modyfikacji wybranej rezerwacji do 30 minut przed planowanym początkiem seansu.

10. System anuluje wszystkie niepotwierdzone rezerwacje 30 minut przed planowanym początkiem seansu.
11. Klient ma możliwość zakupu biletu(ów) na wybrany seans w wybranym kinie.
12. Klient ma możliwość dokonania zapłaty za zakupione bilety za pośrednictwem zewnętrznego systemu płatności elektronicznych.
13. Klient ma możliwość zwrotu zakupionych biletów do 3 godzin przed planowanym seansem.
14. Klient ma możliwość wyboru miejsc na podstawie widoku sali kinowej.
15. Klient ma możliwość wyboru rodzaju biletu przy wyborze miejsc.
16. Klient ma możliwość przeglądania historii rezerwacji oraz zakupionych biletów.
17. Klient ma możliwość dokonania zakupu biletu(ów) bez potrzeby zakładania konta.
18. Klient ma możliwość dokonania rezerwacji biletu(ów) bez potrzeby zakładania konta.
19. Klient ma możliwość okazania biletu w formacie kodu QR.
20. Klient ma możliwość okazania biletu pomimo braku połączenia z siecią Internet.
21. Pracownik kina ma możliwość modyfikacji podstawowych danych o kinie.
22. Pracownik kina ma możliwość modyfikacji informacji o salach dostępnych w kinie.
23. Pracownik kina ma możliwość modyfikacji repertuaru kina.
24. Pracownik kina ma możliwość potwierdzenia rezerwacji klienta.
25. Pracownik kina ma możliwość anulowania rezerwacji klienta.
26. Pracownik kina ma możliwość dokonania sprzedaży biletów klientom w kasie biletowej.
27. Administrator systemu ma możliwość dodawania, modyfikowania i usuwania informacji o kinach należących do sieci kin.
28. Administrator systemu ma możliwość tworzenia oraz modyfikacji kont użytkowników oraz przydzielania im ról.

Powyższa lista jest pełnym zbiorem wymagań, które powinny się znaleźć w ostatecznej wersji systemu. Prototyp systemu, będący celem pracy, będzie miał zaimplementowane wszystkie funkcjonalności dotyczące roli Klienta tj. wymagania od pierwszego do dwudziestego.

3.3 Wymagania niefunkcjonalne

Zbiór tych wymagań definiuje, jakie wymagania na system mają zostać spełnione, oprócz wymagań funkcjonalnych. Wymagania te głównie dotyczą wydajności, bezpieczeństwa i tym podobnych aspektów.

1. System powinien być dostępny bez przerwy. (Dostępność na poziomie 99,9%)
2. System jest w stanie obsługiwać wiele jednocześnie podłączonych urządzeń.
3. Do poprawnego korzystania ze wszystkich funkcji oferowanych przez aplikację, wymagane jest stałe połączenie internetowe.
4. W celu zapewnienia odpowiedniego poziomu bezpieczeństwa, połączenie między serwerem i klientem ma być szyfrowane.
5. System ma wspierać również mechanizm sesji, jako dodatkowy mechanizm zabezpieczający połączenie.
6. Aplikacja kliencka powinna być dostępna na systemach Android (w wersji 4.4 i wyższej) oraz iOS (w wersji 8.0 i wyższej).
7. Aplikacja kliencka powinna zostać uruchomiona na urządzeniu mobilnym niezależnie od stanu połączenia internetowego.
8. Aplikacja serwerowa powinna móc być uruchomiona na serwerach z systemami rodziny Windows Server (wersja 2012 R2 i wyżej) oraz Linux

3.4 Opis podstawowej architektury systemu

Architektura logiczna

System będzie składał się z trzech aplikacji.

Pierwszą z nich jest aplikacja serwerowa, realizowana w architekturze trójwarstwowej (podział na warstwę prezentacji, logiki biznesowej oraz dostępu do danych). Dzięki takiemu podziałowi, w którym wyższa warstwa nie wie o logice zawartej w niższej warstwie (*black box*) zapewniamy m.in. dużą skalowalność rozwiązania (każdą warstwę możemy skalować według potrzeb) oraz ułatwione testowanie rozwiązania poprzez brak skomplikowanych zależności pomiędzy warstwami.

Operacje na bazie danych aplikacji będą wykonywane w warstwie dostępu do danych z częściowym wykorzystaniem podejścia CQRS (*Command Query Responsibility Segregation*), które zapewni jeszcze lepszą skalowalność rozwiązania poprzez, przede wszystkim oddzielenie operacji zapisu danych od ich odczytu. Więcej szczegółów na temat podejścia CQRS oraz jego implementacji znajduje się w podrozdziale 5.5

Pozostałą część systemu stanowią aplikacje mobilne działająca pod kontrolą systemu Android oraz iOS. Aplikacje te będą implementowane z wykorzystaniem platformy Xamarin. Dzięki temu możemy mówić o aplikacji wieloplatformowej, ponieważ większość kodu aplikacji (głównie chodzi tutaj o logikę zawartą w aplikacji) jest współdzielona. Aby umożliwić współdzielenie kodu, będzie zastosowany wzorec architektoniczny Model-View-ViewModel. Dzięki temu wzorcowi dokonujemy separacji aplikacji na warstwy: widoku (View), logiki interfejsu (ViewModel) oraz modelu danych. Logika interfejsu oraz

modelu danych są współdzielone, natomiast osobne widoki odpowiadają specyfice poszczególnych systemów. Wyseparowane warstwy sprawiają, że kod aplikacji jest o wiele bardziej testowalny oraz łatwiejszy w utrzymaniu. Do przechowywania danych lokalnych aplikacji posługuje baza danych SQLite.

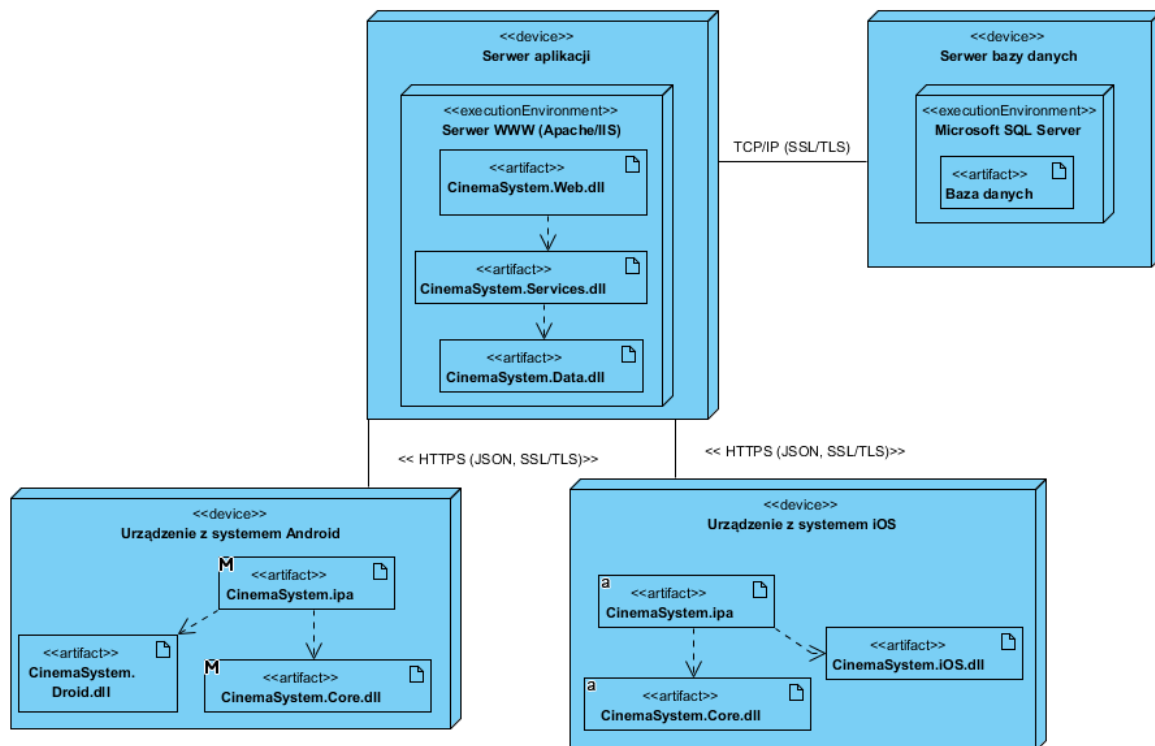
Architektura fizyczna

Aplikacja serwerowa zbudowana będzie w oparciu o framework ASP.NET Core. Będzie mieć ona możliwość pracowania na serwerach z systemami rodziny Windows Server oraz Linux. Wykorzystywany serwer WWW to, w zależności od wybranego systemu, IIS (*Internet Information Services*) oraz Apache.

Aplikacja serwerowa będzie komunikować się z serwerem bazodanowym Microsoft SQL Server znajdującym się na osobnym serwerze (działającym pod kontrolą dystrybucji systemu Windows Server lub Linux), z wykorzystaniem protokołu TLS/SSL w celu zapewnienia odpowiedniego poziomu bezpieczeństwa.

W ramach realizowanego prototypu, serwery aplikacji oraz bazy danych będą hostowane na platformie obliczeniowej Microsoft Azure.

Aplikacje klienckie działające na urządzeniach z systemami Android oraz iOS, będą komunikowały się z serwerem aplikacji, wykorzystując udostępnione przez aplikację *RESTful API*. Formatem przesyłanych danych jest *JSON*. Komunikacja odbywa się za pośrednictwem protokołu HTTPS (ruch jest szyfrowany).



Rysunek 3.1: Architektura fizyczna systemu

Rozdział 4

Projekt aplikacji

4.1 Przypadki użycia

4.2 Interfejs

4.3 Diagram klas

Rozdział 5

Implementacja

5.1 DevOps

5.2 Autoryzacja użytkowników aplikacji

5.3 Synchronizacja danych offline-online

5.4 Bezpieczeństwo aplikacji

5.5 Implementacja wzorca CQRS

5.6 Testy interfejsu aplikacji

Rozdział 6

Podsumowanie

Bibliografia

- [1] *Xamarin Mobile Application Development for iOS*. Packt Publishing, Olton, 2013. ID: 1477484.
- [2] Mark Reynolds. *Xamarin Mobile Application Development for Android*. Packt Publishing, 2014. ID: 48TUR_ejournals2550000001186070.

Załączniki

Spis tabel

Spis rysunków

3.1 Architektura fizyczna systemu 12

Spis listingów

Instrukcja kompilacji i testowego uruchomienia aplikacji