

# System rezerwacji pokoi w hotelu

## Autorzy:

Kazimierz Bagrowski <https://github.com/kazbag>

Michał Kostrzewiński <https://github.com/MikoTheAble>

Filip Kozłowski <https://github.com/fkozlowski>

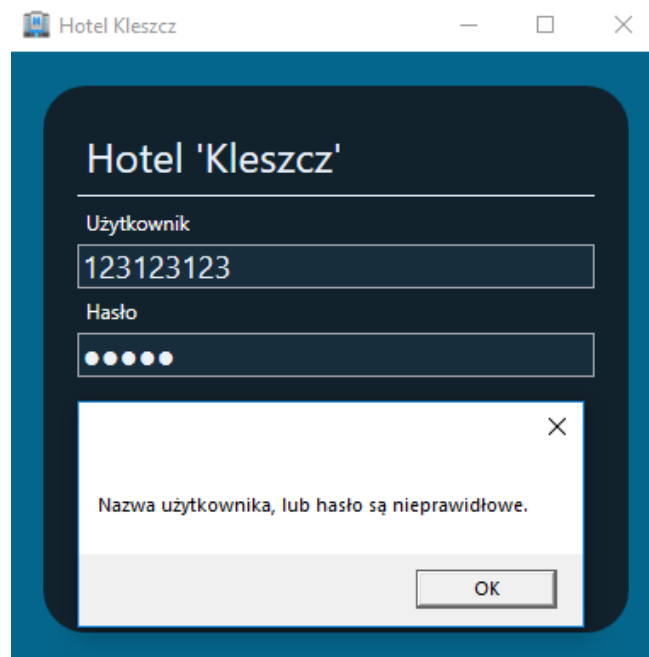
## Ważna informacja

Aby uzyskać dostęp do bazy danych i mieć możliwość zalogowania się, prosimy o kontakt na adres e-mail [filip.kozlowski@microsoft.wsei.edu.pl](mailto:filip.kozlowski@microsoft.wsei.edu.pl) gdyż Azure wymaga ręcznego wpisania adresu IP dla użytkownika.

## Opis projektu:

Aplikacja WPF dla recepcjonisty w hotelu służąca do znalezienia oraz zarezerwowania pokoju dla klienta. Jest ona podpięta pod stworzoną przez nas od zera bazę danych Azure.

Posiada system logowania, który łączy się z w/w bazą i sprawdza, czy dane użytkownika są prawidłowe. Jeżeli nie, to wyświetla odpowiedni **komunikat**:



Jeżeli użytkownik wprowadzi prawidłowe dane, zostaje przeniesiony do okienka **Panel Rezerwacji**.

Panel rezerwacji

Witaj w panelu!

Wyloguj

Kiedy chcesz zarezerwować pokój?

OdDo

luty 2019

pon	wto	śro	czw	pią	sob	nie
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	1	2	3
4	5	6	7	8	9	10

luty 2019

pon	wto	śro	czw	pią	sob	nie
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	1	2	3
4	5	6	7	8	9	10

Dorośli

2

Dzieci do lat 3 (za darmo)

0

Udogodnienia

☐ Pobudka 20zł

☐ Łóżeczko dla dziecka 20zł

☐ Lodówka 20zł

☐ Ekspres do kawy 30zł

☐ Sejf 20zł

☐ Śniadanie do łóżka 50zł na osobę

\* Jeżeli bierzesz pokój z dzieckiem, opcjonalnie weź łóżeczko.

Sprawdź dostępność

W pierwszym kalendarzu „od” użytkownik zaznacza termin, od którego chciałby zarezerwować pokój, natomiast w kalendarzu „do”, do kiedy chciałby zarezerwować. Przeprowadzone zostały testy, które nie przepuszczą użytkownika, jeżeli nie zaznaczy „od” lub „do”, albo jednego z nich. Nie przepuści również, gdy data „do” będzie mniejsza, niż „od”. Nie można również zarezerwować pokoju na 0 dni.

W pokoju musi się znajdować co najmniej 1 osoba dorosła (lub dzieci/młodzież powyżej lat 3) oraz 0 dzieci poniżej lat 3. Podstawowy koszt rezerwacji pokoju wynosi 100zł za dobę dla osoby dorosłej, lub dzieci/młodzieży powyżej lat 3.

Dla dzieci poniżej lat 3 koszt wynosi 0 zł za nocleg. Użytkownik może wybrać udogodnienia takie, jak pobudka, lodówka, sejf, łóżeczko dla dziecka, ekspres do

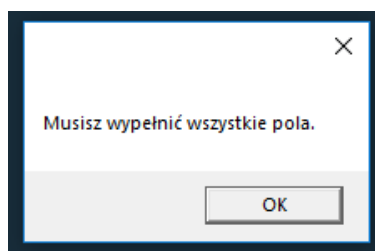
kawy oraz śniadanie do łóżka. Każdy dodatek powoduje dodanie do podstawowej ceny doby hotelowej dodanie odpowiedniej kwoty za dzienne użytkowanie danego urządzenia.

Jeżeli użytkownik wypełni wszystkie pola poprawnie, to aplikacja sprawdzi, czy w danym terminie, np. 24-03-2019 do 29-03-2019 jest jakiś wolny pokój. Jeżeli tak, to pobierze z bazy danych wszystkie informacje o pokoju i wyświetli użytkownikowi w MessageBox'ie liczbę osób (nie uwzględniając dzieci), liczbę dzieci, datę początku, jak i końca rezerwacji, wybrane udogodnienia, jak i całkowity koszt rezerwacji. W kolejnym MessageBox'ie wyświetli informację, ile pokoiów spełnia dane wymagania oraz jeżeli pokój o wprowadzonych „funkcjonalnościach” jest w danym terminie niedostępny, przekaże komunikat, żeby wybrać inny termin rezerwacji, lub zrezygnować z niektórych udogodnień.

Jeżeli pokój będzie dostępny, wyświetli jego numer i aplikacja przeniesie użytkownika do kolejnego okna – rezerwacji. Tutaj poprzez bindowanie do dwóch TextBox'ów zostanie przypisany pierwszy, znaleziony pokój o danych „funkcjonalnościach”, jego numer oraz kwota, którą rezerwujący będzie musiał uiścić za pobyt.



W tym oknie następuje wprowadzenie danych rezerwującego. Formularz jest walidowany poprzez negację „! `String.IsNullOrEmpty()`” dla każdego pola. Użytkownik nie będzie mógł zarezerwować pokoju, jeżeli którekolwiek z pól będzie puste, lub będzie zawierało same białe znaki. Zostanie wyświetlony wtedy stosowny komunikat:



Jeżeli dane będą poprawne, rezerwacja zakończy się sukcesem oraz zostanie przekazane zapytanie do bazy o zarezerwowanie danego pokoju na dane dni.

The screenshot shows a web application interface for room reservations. At the top, there's a header with 'Lista pokoiów' and a 'Wróć' button. Below this is a form with four input fields: 'Imię' (First Name) containing 'Andrzej', 'Telefon' (Phone) containing '666555444', 'Nazwisko' (Last Name) containing 'Łowcaśledzi', and 'Email' containing 'andrzej@sledz.pl'. Below the form, there's a row of three elements: a blue box showing 'Pokój nr 202' and '1110 zł', a button 'Sprawdź, czy cena i numer pokoju się zgadza.', and a 'Rezerwuj' button. At the bottom, a small white dialog box with a close button (X) is open, displaying the message 'Zarezerwowano.' and an 'OK' button.

### Czego nauczyliśmy się podczas pracy nad projektem?

- Pracy w zespole oraz komunikacji
- Podstawową znajomość systemu kontroli wersji
- Podstawową obsługę bazy danych Azure
- Pisać zapytania SQL
- Bindować zmienne
- Szukać ciężkich i niewidocznych na pierwszy rzut oka błędów
- Jak radzić sobie z niechcianymi wyjątkami
- Jak przekazać zmienną z jednego okna do drugiego
- Jak wypić więcej kawy przez tydzień, niż przez ostatnie 5 lat

### Na czym polegaliśmy, co mogliśmy zrobić lepiej?

- Niektóre zmienne są pisane po polsku, natomiast inne po angielsku
- Nie wykorzystaliśmy `async/await`, ponieważ nie znaleźliśmy dla nich zastosowania
- Nie stworzyliśmy testów jednostkowych, ale w samej logice napisaliśmy takowych sporo
- Nie stworzyliśmy biblioteki zawierającej logikę aplikacji
- Zabraliśmy się zbyt późno za tworzenie projektu, przez co nie jest wyszlifowany, lecz zmusił on nas do intensywnej nauki i pozwolił zweryfikować wiedzę z rzeczywistością
- Mogliśmy użyć Entity Framework
- Nie brać dyslektyka do projektu 😊

## Dlaczego akurat ten projekt?

Postanowiliśmy zrobić coś, dzięki czemu nauczymy się oraz wykorzystamy kilka ciekawych technologii, jak np. Azure. Uznaliśmy, że dzięki połączeniu projektu z bazą danych przyswoimy praktyczną wiedzę, którą będziemy mogli później wykorzystać.

## Procentowy udział w projekcie:

Prawdę mówiąc, każdy dawał z siebie 200%. Pracowaliśmy dniami i nocami cały czas dyskutując i pomagając sobie na komunikatorze. Commity również są nieadekwatne co do pracy, gdyż nieraz zdarzało się, że siedzieliśmy razem kilka(naście) godzin nad jednym problemem.

Kazimierz Bagrowski 33%

Filip Kozłowski 33%

Michał Kostrzewiński 33%

Stackoverflow/dokumentacja Microsoft 1%

## Zastosowane technologie:

C#

Azure

SQL

## Struktura bazy danych:

```
CREATE TABLE [dbo].[pracownicy] (  
    [ID] INT IDENTITY (1, 1) NOT NULL,  
    [Username] CHAR (50) NULL,  
    [Password] CHAR (50) NULL,  
    PRIMARY KEY CLUSTERED ([ID] ASC)  
);  
  
CREATE TABLE [dbo].[zamowienia] (  
    [idZamowienia] INT IDENTITY (1, 1) NOT NULL,  
    [idPokoje] INT NOT NULL,  
    [Imie] CHAR (50) NULL,
```

```

[Nazwisko]    CHAR (50) NULL,
[TELEFON]     CHAR (50) NULL,
[Email]       CHAR (50) NULL,
[ReservedSince] DATE    NULL,
[ReservedTo]  DATE     NULL,
[PeopleAmount] INT     NULL,
[WakeUp]      BIT      NULL,
[Fridge]      BIT      NULL,
[Safe]        BIT      NULL,
[ChildBed]    BIT      NULL,
[CoffeeMachine] BIT    NULL,
[BreakfastToBed] BIT    NULL,
[Price]       INT      NULL,
PRIMARY KEY CLUSTERED ([idZamowienia] ASC),
CONSTRAINT [FK_idPokoje] FOREIGN KEY ([idPokoje]) REFERENCES [dbo].[pokoje] ([idPokoje])
);

CREATE TABLE [dbo].[pokoje] (
    [idPokoje]      INT IDENTITY (1, 1) NOT NULL,
    [Liczba_osob]   INT NULL,
    [Budzenie]      BIT NULL,
    [Lodowka]       BIT NULL,
    [Sejf]          BIT NULL,
    [Lozko_dzieciece] BIT NULL,
    [Ekspres_do_kawy] BIT NULL,
    [Sniadanie_do_lozka] BIT NULL,
    PRIMARY KEY CLUSTERED ([idPokoje] ASC)
);

```