

**Dokumentacja projektu**

Przedmiot: **Programowanie**

Tytuł projektu: ***Rezerwacje pokoi program dla recepcji***

Krzysztof Budzik

Nr albumu studenta 60143

Prowadzący: Wykonawca:

*Dr Marek Jaszuk* Krzysztof Budzik Nr albumu 60143

Semestr i symbol kierunku: 4IIZ/2018

Grupa: GP01

Grupa na platformie BB: 03

**Rzeszów 2020**

Spis treści

[**1.** **Opis założeń projektu (wymagania biznesowe)** 3](#_Toc52020276)

[**2.** **Specyfikacja wymagań** 3](#_Toc52020277)

[**1.1** **wymagania funkcjonalne** 3](#_Toc52020278)

[**1.2** **wymagania niefunkcjonalne** 4](#_Toc52020279)

[**3.** **Diagram przypadków użycia** 4](#_Toc52020280)

[**4.** **Harmonogram realizacji projektu** 6](#_Toc52020281)

[**5.** **System kontroli wersji** 6](#_Toc52020282)

[**6.** **Generowanie dokumentacji** 6](#_Toc52020283)

[**7.** **Opis techniczny projektu** 6](#_Toc52020284)

[**8.** **prezentacja warstwy użytkowej projektu** 8](#_Toc52020285)

[**9.** **Raport z testów jednostkowych** 8](#_Toc52020286)

[**10.** **Odnośniki do repozytorium** 11](#_Toc52020287)

[**11.** **Bibliografia** 11](#_Toc52020288)

# **Opis założeń projektu (wymagania biznesowe)**

Recepcja potrzebuje sprawniejszej formy rezerwacji pokoju przy użyciu aplikacji, która umożliwi szybkie wprowadzenie do bazy rezerwowanych pokoi jaki i późniejsze wyszukiwanie zarezerwowanych pokoi. Założeniem firmy jest zrezygnowanie z wersji papierowej i przejście na wersje elektroniczną mającą na celu przyspieszenie pracy pracowników.

Założeniem Projektu jest aplikacja dla moteli i małych hoteli. Ma na celu wspomóc recepcjonistów podczas rezerwacji pokoju, który ułatwi im szybko i prosto rezerwować pokoje. Założeniem projektu jest usprawnienie pracy recepcji i lepsza kontrola nad zarezerwowanymi pokojami. Program będzie posiadać prosty interfejs graficzny.

# **Specyfikacja wymagań**

## **wymagania funkcjonalne**

Wymaganie 2.1.1

System posiada graficzny interfejs użytkownika, umożliwiający wprowadzenie i odczyt danych z bazy.

Wymaganie 2.1.2

aplikacja powinna działać na systemie operacyjnym Windows 7.

Wymaganie 2.1.3

Użytkownik powinien mieć możliwość szybkiego dodania rezerwacji do bazy danych wraz z możliwością aktualizacji rekordu.

Wymaganie 2.1.4

Użytkownik ma mieć możliwość wprowadzenia takich danych jak: imię, nazwisko, numer telefonu, data rezerwacji od, data rezerwacji do, numer pokoju.

Wymaganie 2.1.5

Użytkownik może mieć możliwość przeszukiwania bazy danych po nazwiskach klientów.

Wymaganie 2.1.6

Aplikacja powinna wyświetlać wprowadzane, zaktualizowanie i wyszukiwane rekordy.

Wymaganie 2.1.7

Program ma posiadać 4 przyciski: zapisz, usuń, zresetuj i wyszukaj oraz przycisk zapisz zmieniający się na aktualizuj przy podjęciu próby aktualizacji wprowadzonego już do bazy rekordu.

Wymaganie 2.1.8

Użytkownik ma mieć możliwość szybkiej modyfikacji wrodzonego rekordu do bazy bez ponownego wpisywania wszytkach danych klienta.

## **wymagania niefunkcjonalne**

Wymaganie 2.2.1

Interfejs graficzny programu powinien być intuicyjny dla użytkownika.

Wymaganie 2.2.1

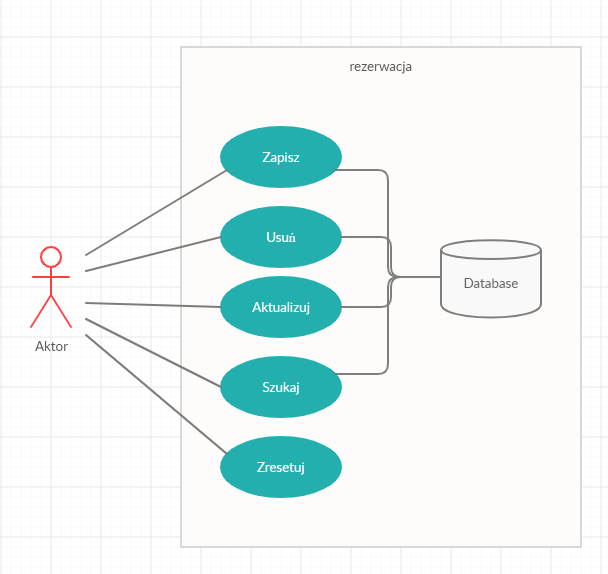
Program działa sprawne

Wymaganie 2.2.2

Program powinien uniemożliwić wprowadzenia już zarezerwowanego pokoju

# **Diagram przypadków użycia**

Diagram przypadku użycia dla aplikacji do rezerwacji opracowany w programie Creately.



Rysunek . Diagram przypadków użycia, źródło: opracowanie własne

W przedstawionym diagramie aktorem jest pracownik, który prezentuje przypadki użycia założeń funkcjonalnych. Za pomocą „Zapisz” Aktor ma możliwość wprowadzenie do bazy danych nowego rekordu z zawartymi informacjami na temat rezerwacji pokoju. Przy użyciu „Usuń” usuwa interesujący go rekord z bazy danych. Przy chęci modyfikacji danych aktor wyciąga podwójnym kliknięciem z wyświetlanej bazy rekord co umożliwia mu zaktualizowanie interesującej go danej. Aktor ma możliwość przeskakiwanie bazy danych po nazwisku przy pomocą przycisku „szukaj”. Natomiast po użyciu „Zresetuj” aktor usuwa wprowadzenie własnoręcznie dane z komórek tekstowych.

# **Harmonogram realizacji projektu**

Rysunek Wykres Gantta, źródło: opracowanie własne

Diagram Gantta opracowany został przy użyciu Microsoft Excel.

# **System kontroli wersji**

System kontroli wersji został wykorzystany GitHub. Jest to serwis internetowy umożlwiający wysłanie projektu programistycznego, aby muc skorzystać z kontroli wersji Git.

# **Generowanie dokumentacji**

Do wygenerowanie dokumentacji końcowej użyto programu Doxygen



Rysunek screen wygenerowanej dokumentacji źródło opracowanie własne

# **Opis techniczny projektu**

Do projektu zostały wykorzystane takie aplikacje jak:

Visual Studio – przy użyciu języka C# ora używając szablonu aplikacji okienkowej WinForms

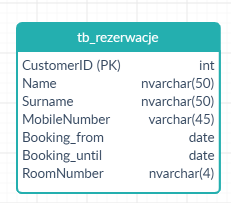
https://creately.com/ - diagram przypadków użycia

Microsoft Excel – diagram Gantta

Sourcetree- sprawna kontrola wersji

Doxygen – komentarze dokumentujące

Baza danych zawiera tabele tb\_rezerwacje która posiada takie atrybuty jak: CustomerID, Name, Surname, mobileNumber, Booking\_from Booking\_until RoomNumber

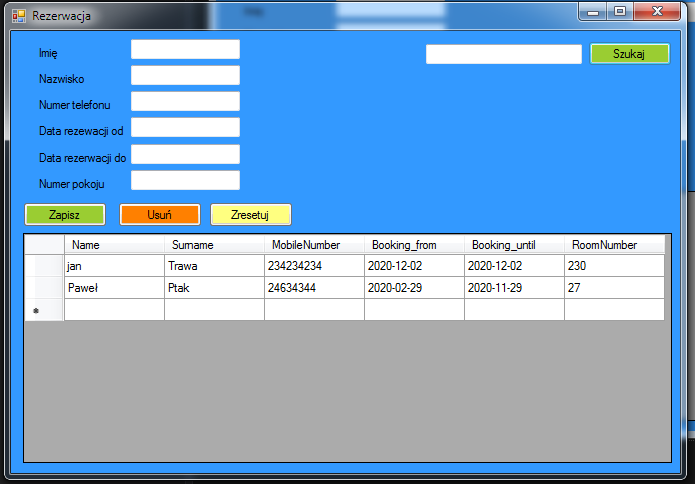


Rysunek diagram tabeli tb\_rezerwacje źródło: opracowanie własne

Głównym plikiem jest Form1.cs w którym są metody:

* btnSave\_Click- metoda jest uruchomiana po naciśnięciu przycisku „Zapisz”. Najpierw sprawdza czy rezerwowany pokój na dany termin nie jest już czasem jest zajęty, jeśli był zajęty to poinformuje aktora komunikatem, że "W podanym terminie dany pokój jest zajęty!". W przeciwnym wypadku wprowadzone dane zostaną wprowadzone do bazy danych jako nowy rekord zawierający wszytki informacje o rezerwacji wprowadzone przez aktora. Wewnątrz tej metody jest aktualizacja rekordu. w przypadku gdy przycisk „zapisz” zmieni swoja wartość na „zaktualizuj” będzie możliwe aktualizacją danego rekordu a będzie to możliwe, jeśli zostanie aktywowana metoda „dataGridView\_DoubleClick”
* btnSearch\_Click-metoda uruchamiana za pomocą przycisku „szukaj” po przekazaniu nazwiska do funkcji FillDataGridView która przeszukuje tabele czy nie ma rezerwacji dana osoba i jeśli znajdzie rekord to wyświetli p przygotowanym okienku wszystkie informacje na temat rezerwacji danej osoby.
* dataGridView\_DoubleClick- metoda, która zadziała po podwójnym kliknięciu w rekord i wpisze w komórki testowe umożliwiając aktorowi szybką i sprawna aktualizacje.
* btnReset\_Click- uruchomiana jest po naciśnięciu przycisku reset. Odwołuje się do funkcji reset która czyści wszystkie wprowadzone dane do komórki tekstowe oraz zamienia przekazową wartość przycisku btnSave na „Zapisz”
* btnDelete\_Clic- uruchamiana za pomocą przycisku „usuń” powoduje przekazanie wartości wprowadzonej Customer\_ID procedury w bazie danych która jest odpowiedzialna za usuniecie danego rekordu.

# **prezentacja warstwy użytkowej projektu**

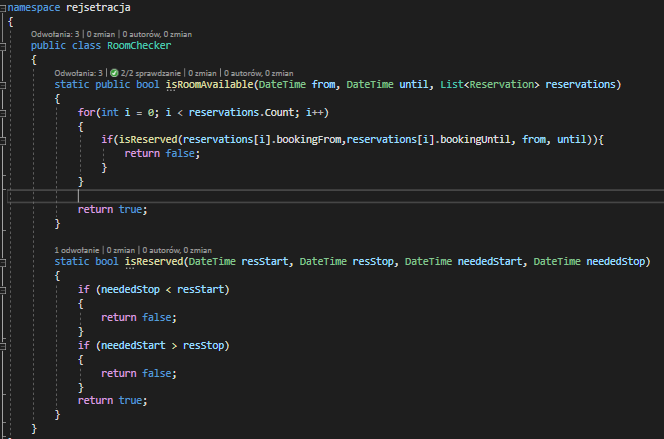


Rysunek widok programu rezerwacji źródło: opracowanie własne

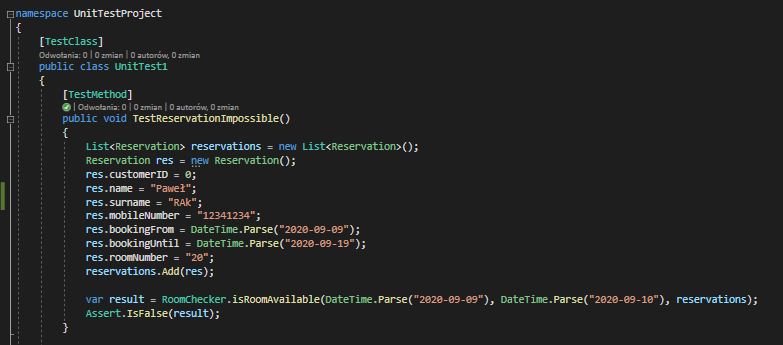
Program posiada prostą nawigację co pozwala na intuicyjną obsługę. Umożliwia szybkie i sprawne przeszukiwanie i wprowadzenie danych do tabeli. Co powinno znacznie usprawnić pracę w recepcji. Dwukrotne kliknięcie w rekord umożliwia automatyczne wypełnienie się pól danych w celu aktualizacji danego rekordu. Przyciski, opisy oraz alerty są w języku Polskim.

# **Raport z testów jednostkowych**

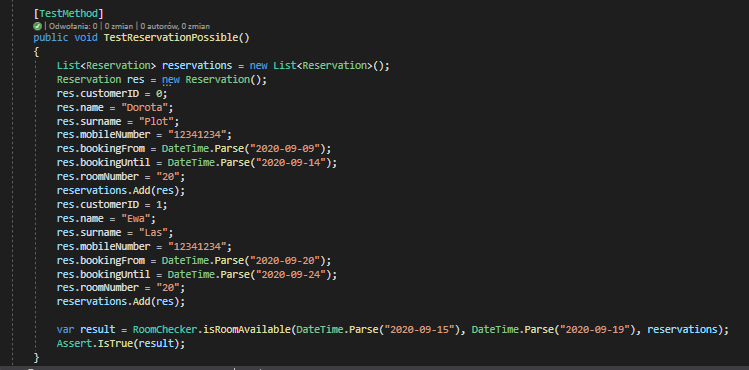
Zostały stworzone dwa testy jednostkowe mające na celu sprawdzenie poprawność wprowadzanych danych. dwa testy zakończyły wynikiem pozytywnym. Do pierwszej metody testującej „TestReservationImpossible” zostały przekazane takie dane które nie powinny być wprowadzone do bazy, ponieważ próbujemy zarezerwować pokój, który jest w tym czasie zarezerwowany, oczekiwaliśmy, że po przekazaniu do funkcji „isRoomAvailable” zwróci nam „false” i taki wynik nam ta funkcja zwróciła, dlatego wynik testu pozytywny. Funkcja zadziałała tak ze jeśli rozważamy początek rezerwacji pierwszej przesłanej daty, ale nie jest późniejsza niż jej koniec drugiej daty więc pierwszy warunek IF nic nie zwróci, ale początek drugiej daty nie jest późniejszy niż końcowa data rezerwacji pierwszej, dlatego funkcja „isReserved” zwróci „false” i funkcja „isRoomAvailable” również zwróci „false” czyli taki wynik jak się spodziewaliśmy. W drugim teście jednostkowym podjęto próbę wprowadzenia rezerwacji pokoju, który nie jest w tym terminie zarezerwowany również rezultatem funkcji byłą prawda, czyli oznaczało to, że można zarezerwować pokój w tej dacie, ponieważ nie jest on zarezerwowany



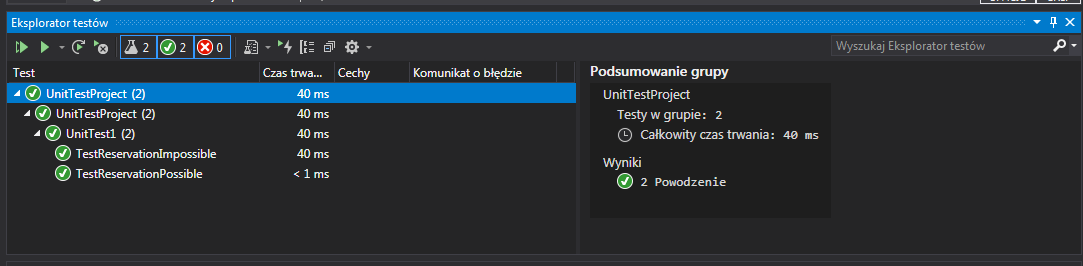
Rysunek metoda sprawdzający czy nie rezerwujemy pokoju o tej samej dacie rummchecker.cs źródło: opracowanie własne



Rysunek pierwszy test jednostkowy źródło: opracowanie własne



Rysunek drugi test jednostkowy źródło: opracowanie własne



Rysunek 8 wynik testów jednostkowych źródło: opracowanie własne

# **Odnośniki do repozytorium**

[Doxygen](https://megawrzuta.pl/download/41e461281d7ebf86ab189cf2cd584db6.html)

[Repozytorium](https://github.com/hoyt90-3/Project_Room-reservation)

# **Bibliografia**

[1] <https://docs.microsoft.com/pl-pl/dotnet/csharp/tour-of-csharp/>

[2] <https://pl.wikipedia.org/wiki/Diagram_Gantta>

[3] <https://creately.com/>