

**WYDZIAŁ MATEMATYKI I INFORMATYKI  
UNIwersytetu Łódzkiego**

Kierunek studiów: Informatyka

Specjalizacja: Grafika Komputerowa i Projektowanie Gier

**Aplikacja internetowa „System zgłaszania błędów”**

Mikołaj Szymański

Numer albumu: 359892

Praca licencjacka napisana  
w Katedrze Funkcji Rzeczywistych  
pod kierunkiem  
dra Sebastiana Lindnera

**Łódź 2018**

## Spis treści

Wstęp .....	3
1. Wzorce architektury używane w projektowaniu aplikacji. ....	3
1.1. Opis wzorca architektury MVC .....	3
1.2. Opis wzorca architektury MVP .....	4
1.3. Opis wzorca architektury MVVM .....	4
1.4. Który wzorec jest odpowiedni dla mojej aplikacji?.....	4
2. Dokumentacja użytkownika .....	5
2.1. Do czego służy aplikacja .....	5
2.2. Instrukcja obsługi aplikacji.....	5
2.2.2. Tworzenie zgłoszenia.....	6
2.2.3. Przeglądanie zgłoszeń.....	7
2.2.4. Edycja zgłoszeń.....	9
3. Dokumentacja programisty .....	13
3.2. Użyte narzędzia programistyczne.....	13
3.2.1. Instalacja.....	13
3.2.2. Konfiguracja.....	14
3.3. Opis działania kodu .....	16
3.4. Klasy.....	16
3.4.1. Klasa Account .....	16
3.4.2. Klasa Ticket .....	16
3.4.3. Klasa FileUploadHelperExtension.....	16
3.5. Opis działania wybranych funkcji .....	17
3.5.1. Funkcja DownloadFile.....	17
3.6. Graficzna reprezentacja schematu bazy danych.....	17
4. Podsumowanie .....	17
5. Bibliografia.....	17

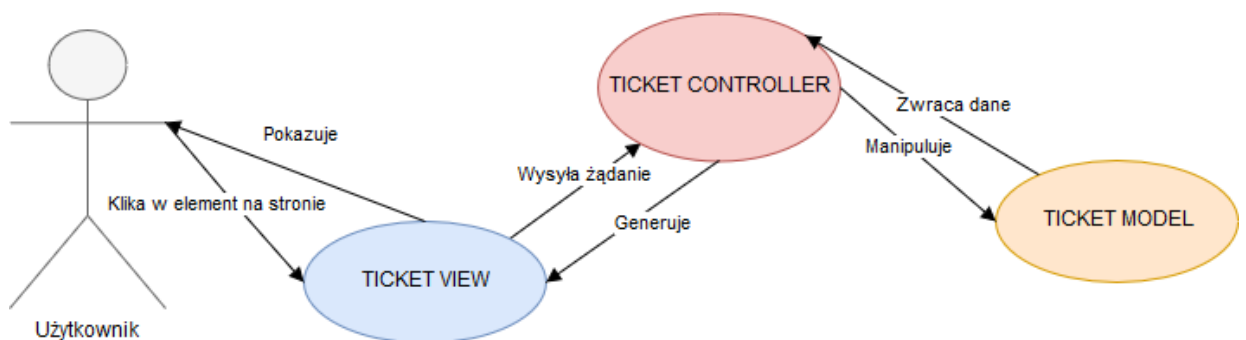
# Wstęp

Tu będzie wstęp.

## 1. Wzorce architektury używane w projektowaniu aplikacji.

### 1.1. Opis wzorca architektury MVC

Wzorzec Model-View-Controller jest przeznaczony do projektowania aplikacji internetowych. **Model** reprezentuje naszą warstwę biznesową, czyli nie tylko klasy mapujące dane pobrane z bazy danych. To również funkcjonalności potrzebne do działania naszej aplikacji. **View** jest odpowiedzialny za prezentowanie użytkownikowi danych, pobieranych z Modelu, który jest dla niego tylko do odczytu. **Controller** przekierowuje wszelkie żądania użytkownika na wywołanie metod konkretnego **Modelu**, oraz przygotowuje dane dla **View**, z żądaniem jego pokazania. Każdy kontroler posiada bezpośrednie wskazania na określone modele i widoki, z którymi współpracuje, a jednocześnie w aplikacji może istnieć wiele kontrolerów. W danym momencie tylko jeden z nich steruje aplikacją.<sup>1</sup> Przykładem użycia tej architektury w mojej pracy, jest tworzenie nowego zgłoszenia. Użytkownik poprzez kliknięcie odpowiedniego przycisku na stronie, wysyła żądanie stworzenia nowego zgłoszenia do **Ticket Controller'a**. Następnie manipulując **Ticket Modelem**, uzyskuje konkretne dane, które później może modyfikować, a na końcu generuje końcowy **View**.



Schemat działania **Model-View-Controller**

<sup>1</sup> <https://pl.wikipedia.org/wiki/Model-View-Controller>

### 1.2.Opis wzorca architektury MVP

Wzorzec Model-View-Presenter używany jest głównie do projektowania aplikacji mobilnych. **Presenter** w tym wzorcu ma podobne zastosowanie co kontroler w MVC. Dodatkowo posiada on w sobie logikę biznesową, która w MVC jest pod Modelem. Tutaj dane nie są przekazywane bezpośrednio z **Model** na **View**, ale **presenter** wykonuje zapytania do modelu o konkretne wartości. Później są one przetwarzane i wysyłane na **View**. **Model** jest zwykłą reprezentacją danych, zawierająca powiązania i struktury danych. **View** ma to samo zadanie co w architekturze MVC, czyli prezentuje użytkownikowi dane pobrane z **Modelu**. W tej architekturze, presenter odnosi się do jednego widoku, nie może obsługiwać wielu.

### 1.3.Opis wzorca architektury MVVM

Wzorzec Model-View-ViewModel to wzorzec wykorzystywany głównie do aplikacji graficznych, w których dane są mocno ze sobą powiązane. Powstał na bazie architektury MVC, i stąd **Model** jest niezmienny w swoich zastosowaniach. **View** jest interfejsem użytkownika, który nie ma pojęcia o **ViewModel-u**. **ViewModel** pobiera dane z Modelu i przygotowuje je do wyświetlenia na możliwie wiele Widoków. Ten wzorzec nie ogranicza się do ilości obsługiwanych widoków.

### 1.4.Który wzorzec jest odpowiedni dla mojej aplikacji?

Każdy z opisanych wzorców ma swoje wady i zalety. Zastosowanie **MVP** ogranicza nas do obsługi tylko jednego widoku, przy użyciu **MVC** bądź **MVVM** nie musimy się martwić o limit obsługi widoków. Jednak w porównaniu do **MVVM**, popularność wśród aplikacji internetowych architektury **MVC** jest zauważalnie większa. Użycie tej architektury pomaga programistom współpracować ze sobą nad jednym kodem. Dzięki modułowej budowie (Model – View- Controller) modyfikacja kodu jest prosta a sam kod jest przejrzystszy.

## 2. Dokumentacja użytkownika

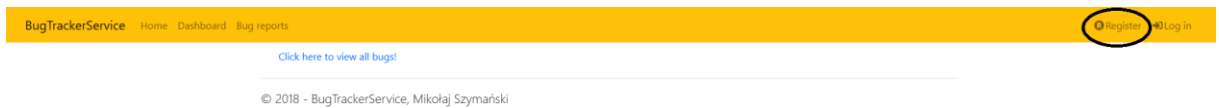
### 2.1. Do czego służy aplikacja

Aplikacja została stworzona na potrzeby firm informatycznych. Jej celem jest umożliwienie kontrolowania ilości błędów w programach danej firmy. Użytkownik, czyli osoba posiadająca do własnego użytku konkretny program, po zarejestrowaniu się na stronę, posiada możliwość zgłoszenia problemu wynikłego podczas korzystania z programu komputerowego. Podając szczegóły, wrzucając zdjęcia obrazujące powstały problem, użytkownik zaznacza serwisantów z powstałym kłopotem. Serwisanci po przeanalizowaniu problemu, podejmują po swojej stronie odpowiednie kroki, aby usunąć awarię i przywrócić funkcjonalność programu.

### 2.2. Instrukcja obsługi aplikacji

#### 2.2.1. Rejestracja użytkownika

Użytkownik, który posiada oprogramowanie wystawione przez konkretną firmę, może zarejestrować się na stronę przez kliknięcie w górnym prawym rogu **Register**.

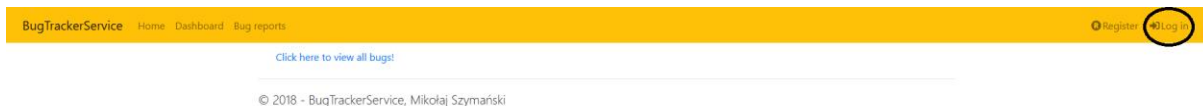


#### Instrukcja rejestracji etap 1

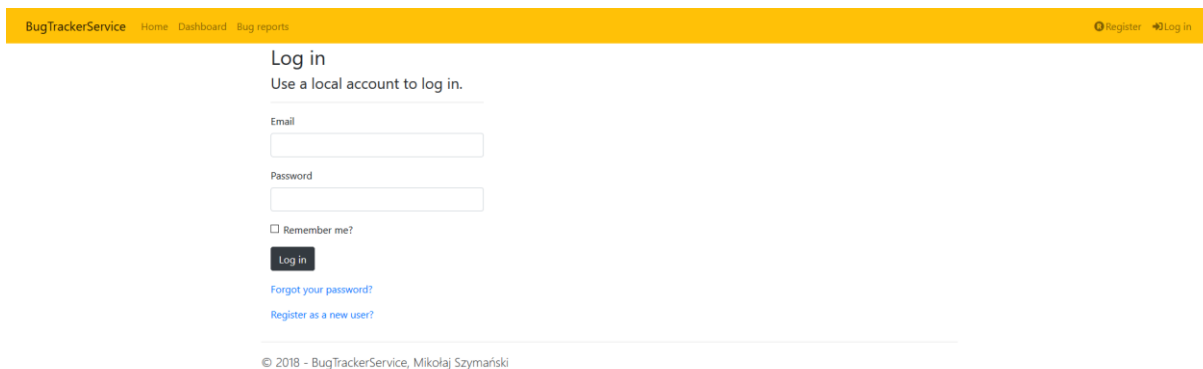
Po ukazaniu się formularza rejestracji, użytkownik podaje swoje dane, czyli imię i nazwisko, swojego e-maila potrzebnego do potwierdzenia (bez tego nie można korzystać z aplikacji), swoje hasło oraz jego potwierdzenie składające się z przynajmniej 8 znaków w tym: jedna wielka litera oraz znak specjalny, nazwa firmy, w której pracuje oraz telefon kontaktowy.

#### Instrukcja rejestracji etap 2

Po kliknięciu przycisku **Register** na dole formularza, użytkownik dostaje e-maila z linkiem do potwierdzenia swojej tożsamości. Po pozytywnej weryfikacji, użytkownik może zalogować się na stronę klikając w prawym górnym rogu ikonkę **Log in**. Podaje w formularzu swój login i hasło podane podczas rejestracji i cieszy się z możliwości oferowanych przez aplikację.



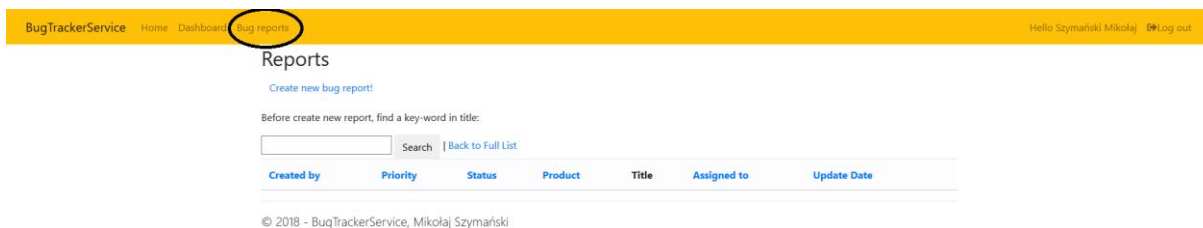
#### Instrukcja logowania etap 1



#### Instrukcja logowania etap 2

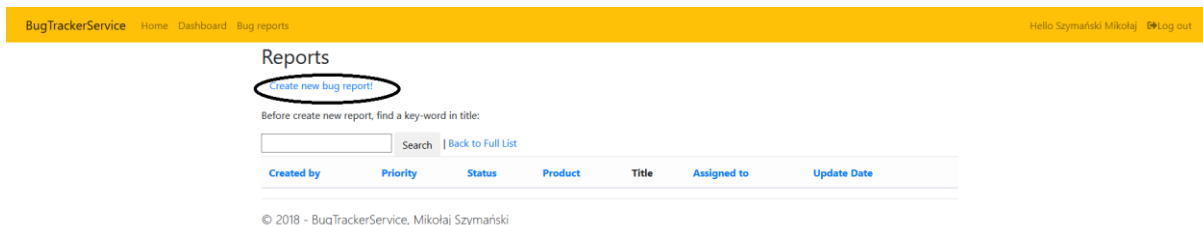
### 2.2.2. Tworzenie zgłoszenia

Aby móc stworzyć zgłoszenie, użytkownik musi być zalogowany do serwisu. Na górnym pasku strony kliknąć w zakładkę **Bug Reports**.



Wejście w zakładkę Bug Reports

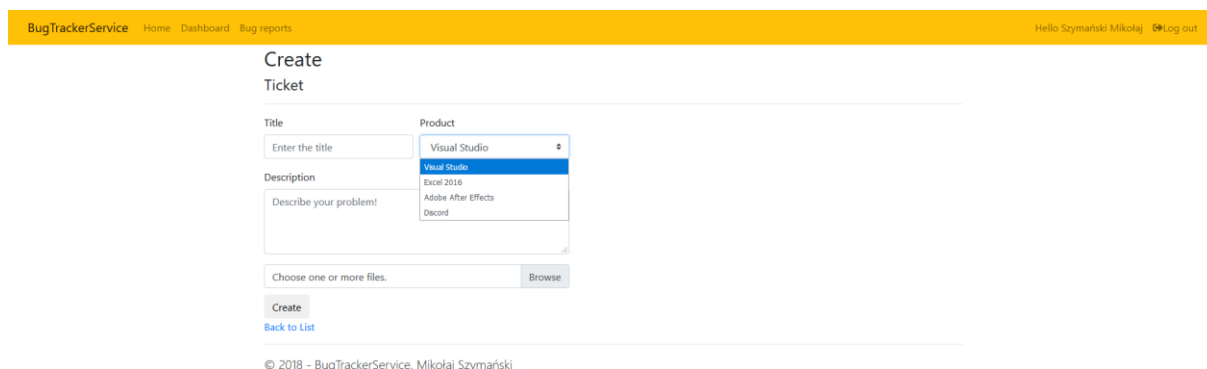
Po kliknięciu w **Create new bug report!** otworzy nam się prosty formularz do zgłoszenia problemu napotkanego podczas korzystania z jednej z aplikacji.



Wejście do formularza tworzenia zgłoszenia

Podajemy w nim tytuł (tak, żeby się dobrze kojarzył z danym problemem), jego szczegółowy opis, wybieramy jakiego oprogramowania się problem dotyczy, i jeżeli jest możliwość, to pokazać zdjęcia obrazujące dany problem.

Po kliknięciu przycisku **Create** nasze zgłoszenie zostaje utworzone i każdy kto jest zalogowany może je zobaczyć.



BugTrackerService Home Dashboard Bug reports Hello Szymański Mikołaj Log out

### Create Ticket

Title:

Product:

Description:

Choose one or more files.

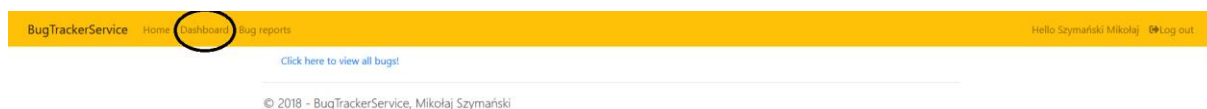
[Back to List](#)

© 2018 - BugTrackerService, Mikołaj Szymański

Formularz tworzenia zgłoszenia

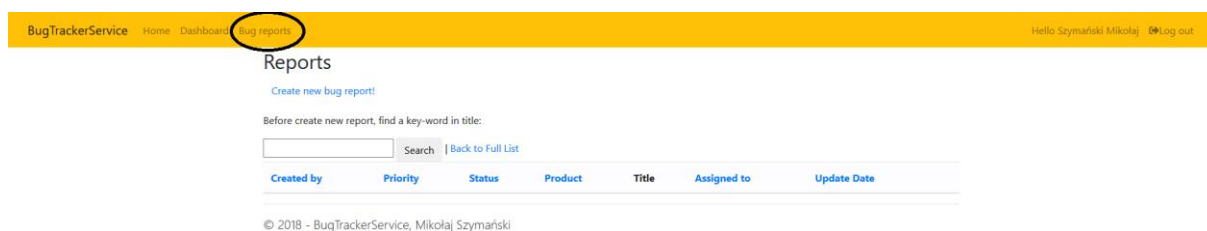
### 2.2.3. Przeglądanie zgłoszeń

Mamy możliwość przeglądania zgłoszeń swoich, oraz wszystkich. Po kliknięciu w **Dashboard** ukazują nam się wszystkie zgłoszenia stworzone tylko i wyłącznie przez nas. Jeżeli jesteśmy serwisantem, pokażą nam się wszystkie zgłoszenia do których jesteśmy przypisani.



Wejście w zakładkę **Dashboard**

Natomiast jeżeli klikniemy w ikonkę **Bug Reports**, to ukazać nam się wszystkie zgłoszenia.



Wejście w zakładkę **Bug Reports**

Tabela pokazuje nam **przez** kogo został stworzony problem, jego priorytet (od niskiego po wysoki), jego status (od oczekującego do skończonego), jakiego produktu się tyczy, jego tytuł, do kogo jest przypisany i jego data ostatniej aktualizacji.

## a) Przegląd wszystkich zgłoszeń

The screenshot shows the 'Reports' page of the BugTrackerService. At the top, there is a navigation bar with 'BugTrackerService', 'Home', 'Dashboard', and 'Bug reports'. On the right, it says 'Hello Szymański Mikołaj' and 'Log out'. Below the navigation bar, the page title is 'Reports'. There is a link 'Create new bug report!' and a search bar with the text 'Before create new report, find a key-word in title:'. Below the search bar, there is a table with columns: 'Created by', 'Priority', 'Status', 'Product', 'Title', 'Assigned to', and 'Update Date'. The table contains two rows of data. The first row is for 'Admin Admini' with 'Low' priority, 'In Queue' status, 'Excel 2016' product, and title 'Tabela sie nie podsumowuje'. The second row is for 'Szymański Mikołaj' with 'Low' priority, 'In Queue' status, 'Visual Studio' product, and title 'Nie mogę utworzyć nowej tabelki.'. Each row has a 'Show details' link. At the bottom, there is a copyright notice: '© 2018 - BugTrackerService, Mikołaj Szymański'.

Created by	Priority	Status	Product	Title	Assigned to	Update Date	
Admin Admini	Low	In Queue	Excel 2016	Tabela sie nie podsumowuje	Not Assigned	sobota, 5 maja 2018 00:20	Show details
Szymański Mikołaj	Low	In Queue	Visual Studio	Nie mogę utworzyć nowej tabelki.	Pracownik Pracownik	sobota, 5 maja 2018 00:21	Show details

Zgłoszenia widoczne przez każdego

## b) Przegląd zgłoszeń stworzonych przez użytkownika

The screenshot shows the 'Your Tickets' page of the BugTrackerService. The navigation bar is the same as in the previous screenshot. The page title is 'Your Tickets'. Below the title, there is a table with columns: 'Created by', 'Priority', 'Status', 'Product', 'Title', 'Assigned to', and 'Update Date'. The table contains one row of data for 'Szymański Mikołaj' with 'Low' priority, 'In Queue' status, 'Visual Studio' product, and title 'Nie mogę utworzyć nowej tabelki.'. The 'Created by' column is circled. At the bottom, there is a copyright notice: '© 2018 - BugTrackerService, Mikołaj Szymański'.

Created by	Priority	Status	Product	Title	Assigned to	Update Date	
Szymański Mikołaj	Low	In Queue	Visual Studio	Nie mogę utworzyć nowej tabelki.	Pracownik Pracownik	sobota, 5 maja 2018 00:21	Show details

Zgłoszenia stworzone przez użytkownika

## c) Przegląd zgłoszeń przypisanych do pracownika

The screenshot shows the 'Your Tickets' page of the BugTrackerService. The navigation bar is the same as in the previous screenshots. The page title is 'Your Tickets'. Below the title, there is a table with columns: 'Created by', 'Priority', 'Status', 'Product', 'Title', 'Assigned to', and 'Update Date'. The table contains one row of data for 'Szymański Mikołaj' with 'Low' priority, 'In Queue' status, 'Visual Studio' product, and title 'Nie mogę utworzyć nowej tabelki.'. The 'Assigned to' column is circled. At the bottom, there is a copyright notice: '© 2018 - BugTrackerService, Mikołaj Szymański'.

Created by	Priority	Status	Product	Title	Assigned to	Update Date	
Szymański Mikołaj	Low	In Queue	Visual Studio	Nie mogę utworzyć nowej tabelki.	Pracownik Pracownik	sobota, 5 maja 2018 00:21	Show details

Zgłoszenia przypisane do pracownika

Po kliknięciu w opcję **Show details** ukaże nam się cały opis naszego zgłoszenia. Kto jest przypisany do danego zgłoszenia, jego status oraz priorytet. Możemy również dodawać komentarze do zgłoszenia, co jest szybką formą komunikacji między użytkownikiem a serwisantem.



BugTrackerService Home Dashboard Bug reports Hello Szymański Mikołaj Log out

## Details

### Ticket

**Created by**  
Szymański Mikołaj

**Assigned to**  
Pracownik Pracownik

**Product**  
Visual Studio

**Title**  
Nie mogę utworzyć nowej tabelki.

**Description**  
Opis tabelki.

**Status**  
In Queue

**Priority**  
Low

**Create Date**  
sobota, 5 maja 2018 00:20

**Update Date**  
sobota, 5 maja 2018 00:21

**Comments**

Comment

Add comment!

Add comment

[Edit](#)  
[Back to List](#)

© 2018 - BugTrackerService, Mikołaj Szymański

Widok szczegółowy zgłoszenia

## 2.2.4. Edycja zgłoszeń

Jeżeli jakaś rzecz w naszym zgłoszeniu jest zła (zły opis) bądź brakuje pewnych zdjęć do zobrazowania problemu dokładniej, dobrą opcją jest edycja takiego zgłoszenia. Drugą możliwością edycji zgłoszeń jest przypisanie się serwisanta do konkretnego zgłoszenia. Serwisanta może przypisać również główny administrator. Aby dostać się do edycji zgłoszenia, trzeba wejść przez opcję **Show details**, a następnie na dole zgłoszenia kliknąć **Edit**.

BugTrackerService Home Dashboard Bug reports Hello Szymański Mikołaj Log out

## Details

### Ticket

**Created by**  
Szymański Mikołaj

**Assigned to**  
Pracownik Pracownik

**Product**  
Visual Studio

**Title**  
Nie mogę utworzyć nowej tabelki.

**Description**  
Opis tabelki.

**Status**  
In Queue

**Priority**  
Low

**Create Date**  
sobota, 5 maja 2018 00:20

**Update Date**  
sobota, 5 maja 2018 00:21

**Comments**

Comment

Add comment!

Add comment

[Edit](#)  
[Back to List](#)

© 2018 - BugTrackerService, Mikołaj Szymański

Wejście w edycję zgłoszenia

## a) Edycja przez serwisanta

The screenshot shows the 'Edit Ticket' form in the BugTrackerService application. The top navigation bar is yellow and contains 'BugTrackerService', 'Home', 'Dashboard', 'Bug reports', and a user greeting 'Hello Pracownik Pracownik' with a 'Log out' link. The form itself is titled 'Edit Ticket' and includes a checkbox 'Do you want assign? Product' with a 'Visual Sta' dropdown. Below this are input fields for 'Title' (containing 'Nie moge utworzyc nowej tabelki.') and 'Description' (containing 'Opis tabelki.'). There are also dropdowns for 'Status' (set to 'In Queue') and 'Priority' (set to 'Low'). A file upload section says 'Choose one or more files.' with a 'Browse' button. At the bottom of the form are a blue 'Submit' button and a 'Back to List' link. The footer of the form area says '© 2018 - BugTrackerService, Mikołaj Szymański'.

Formularz edycji zgłoszenia przez serwisanta

## b) Edycja przez administratora

This screenshot shows the 'Edit Ticket' form from the administrator's perspective. The top navigation bar is yellow and includes 'BugTrackerService', 'Home', 'Dashboard', 'Bug reports', and 'Administrator panel'. The user greeting is 'Hello Admin Admin' with a 'Log out' link. The form structure is identical to the one for the service technician, with fields for 'Title', 'Description', 'Status', 'Priority', and file upload. The 'Employee' dropdown is now set to 'Pracownik'. The footer of the form area remains '© 2018 - BugTrackerService, Mikołaj Szymański'.

Formularz edycji zgłoszenia przez administratora

### 2.2.5. Panel administratora

Będąc głównym administratorem serwisu, mamy dostęp do **Panelu administratora**, który umożliwia nam dodanie Produktów (oprogramowania) oraz dodawania pracowników, usuwania użytkowników oraz pracowników.

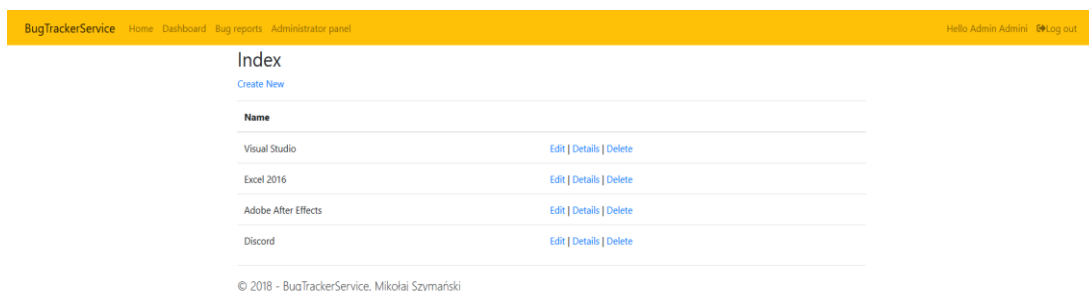
The screenshot shows the 'Administrator Panel' in the BugTrackerService application. The top navigation bar is yellow and contains 'BugTrackerService', 'Home', 'Dashboard', 'Bug reports', and 'Administrator panel'. The user greeting is 'Hello Admin Admin' with a 'Log out' link. The main content area is titled 'Administrator Panel' and contains two links: 'Products' and 'Users'. The footer of the page says '© 2018 - BugTrackerService, Mikołaj Szymański'.

Panel administratora

## a) Zakładka **Products**

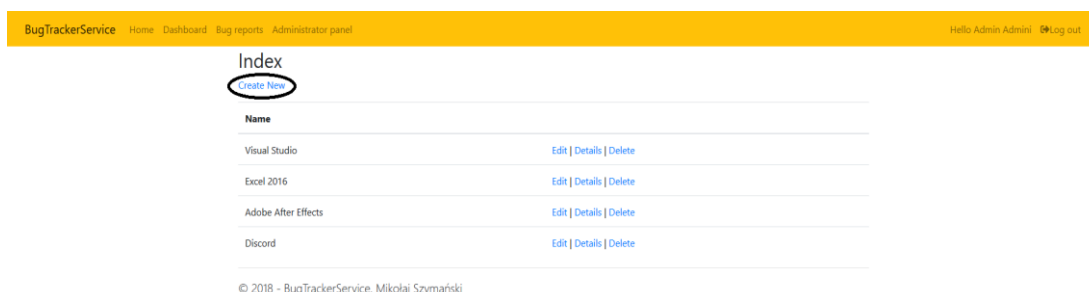
### a. Przegląd

Po kliknięciu w opcję **Products** możemy edytować nasz produkt (zmienić jego nazwę),



Lista produktów

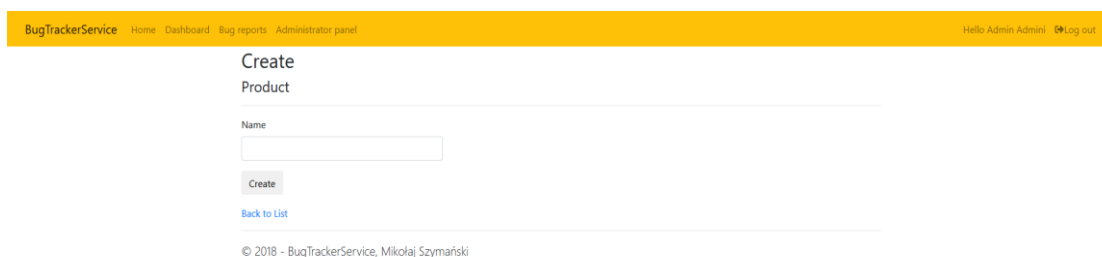
usunąć go z listy produktów oraz dodać nowy (po kliknięciu **Create New**).



Wejście w dodanie nowego produktu

### b. Tworzenie

Po kliknięciu w opcję **Create new** możemy dodać nowy produkt do naszej listy, wpisując jego nazwę i zatwierdzając przyciskiem **Create**.



Formularz dodania nowego produktu

## b) Zakładka **Users**

### a. Przegląd

Otwierając zakładkę **Users** ukazuje nam się lista wszystkich zarejestrowanych użytkowników, oraz ich podstawowe dane podawane podczas rejestracji.

BugTrackerService Home Dashboard Bug reports Administrator panel Hello Admin Admin! Log out							
Index							
<a href="#">Create new Employee!</a>							
FirstName	LastName	CompanyName	WorkerCardNumber	UserName	Email	EmailConfirmed	PhoneNumber
Mikołaj	Szymański	MikołajCompany				<input checked="" type="checkbox"/>	555
						<a href="#">Edit</a>   <a href="#">Delete</a>	
Pracownik	Pracownik	Pracownik	010	pracownik@pracownik.pl	pracownik@pracownik.pl	<input checked="" type="checkbox"/>	010
						<a href="#">Edit</a>   <a href="#">Delete</a>	
Admini	Admin	Administracja	00-00	admin@admin.com	admin@admin.com	<input checked="" type="checkbox"/>	00-00
						<a href="#">Edit</a>   <a href="#">Delete</a>	
© 2018 - BugTrackerService, Mikołaj Szymański							

Lista użytkowników

### b. Tworzenie

Chcąc dodać nowego pracownika należy kliknąć w **Create new Employee**, a link przekieruje nas na stronę do dobrze znanego nam formularza rejestracji nowego użytkownika (2.2.1)

BugTrackerService Home Dashboard Bug reports Administrator panel Hello Admin Admin! Log out							
Index							
<a href="#">Create new Employee!</a>							
FirstName	LastName	CompanyName	WorkerCardNumber	UserName	Email	EmailConfirmed	PhoneNumber
Mikołaj	Szymański	MikołajCompany				<input checked="" type="checkbox"/>	555
						<a href="#">Edit</a>   <a href="#">Delete</a>	
Pracownik	Pracownik	Pracownik	010	pracownik@pracownik.pl	pracownik@pracownik.pl	<input checked="" type="checkbox"/>	010
						<a href="#">Edit</a>   <a href="#">Delete</a>	
Admini	Admin	Administracja	00-00	admin@admin.com	admin@admin.com	<input checked="" type="checkbox"/>	00-00
						<a href="#">Edit</a>   <a href="#">Delete</a>	
© 2018 - BugTrackerService, Mikołaj Szymański							

Wejście w dodanie nowego pracownika

### 3. Dokumentacja programisty

#### 3.2. Użyte narzędzia programistyczne

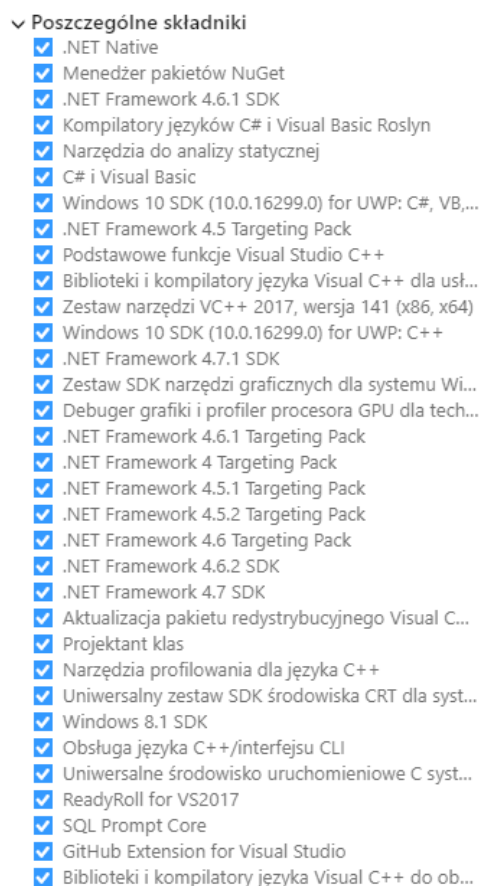
Do stworzenia aplikacji użyłem frameworku ASP.NET Core w wersji 2.0, programu Visual Studio 2017 na licencji studenckiej (<https://www.microsoft.com/net/download/Windows/build>). Do tworzenia bazy danych użyłem Entity Framework, który jest dodatkiem do programu Visual Studio.

Do stworzenia części wizualnej aplikacji użyłem Bootstrapa w wersji 4 (<https://getbootstrap.com/docs/4.1/getting-started/download/>), oraz FontAwesome w wersji darmowej do ikon przy napisach(<https://fontawesome.com/>).

Głównym językiem programowania jest C#.

##### 3.2.1. Instalacja

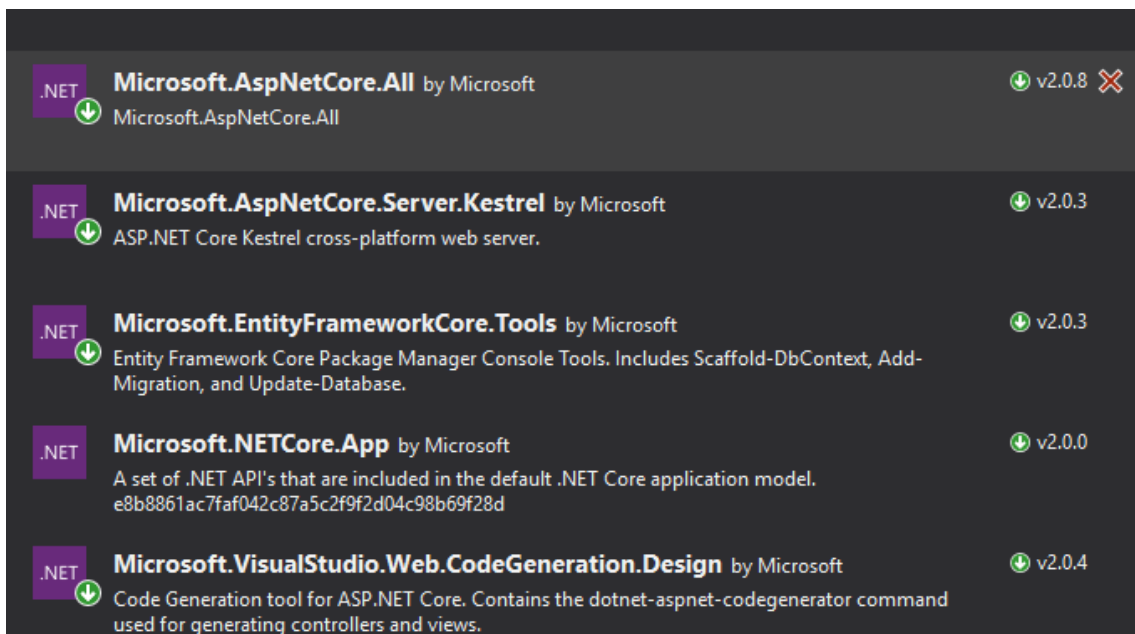
Instalując program Visual Studio trzeba zaznaczyć w pakietach roboczych: **Programowanie aplikacji klasycznych dla platformy .NET, Opracowywanie zawartości dla platformy ASP.NET i sieci Web, Magazynowanie i przetwarzanie danych, Programowanie dla wielu platform w środowisku .NET Core** – kiedy pobierzemy te wszystkie jesteśmy gotowi przystąpić do pracy w ASP.NET Core.

- 
- ▼ Poszczególne składniki
- ☒ .NET Native
  - ☒ Menedżer pakietów NuGet
  - ☒ .NET Framework 4.6.1 SDK
  - ☒ Kompilatory języków C# i Visual Basic Roslyn
  - ☒ Narzędzia do analizy statycznej
  - ☒ C# i Visual Basic
  - ☒ Windows 10 SDK (10.0.16299.0) for UWP: C#, VB,...
  - ☒ .NET Framework 4.5 Targeting Pack
  - ☒ Podstawowe funkcje Visual Studio C++
  - ☒ Biblioteki i kompilatory języka Visual C++ dla usł...
  - ☒ Zestaw narzędzi VC++ 2017, wersja 141 (x86, x64)
  - ☒ Windows 10 SDK (10.0.16299.0) for UWP: C++
  - ☒ .NET Framework 4.7.1 SDK
  - ☒ Zestaw SDK narzędzi graficznych dla systemu Wi...
  - ☒ Debugger grafiki i profiler procesora GPU dla tech...
  - ☒ .NET Framework 4.6.1 Targeting Pack
  - ☒ .NET Framework 4 Targeting Pack
  - ☒ .NET Framework 4.5.1 Targeting Pack
  - ☒ .NET Framework 4.5.2 Targeting Pack
  - ☒ .NET Framework 4.6 Targeting Pack
  - ☒ .NET Framework 4.6.2 SDK
  - ☒ .NET Framework 4.7 SDK
  - ☒ Aktualizacja pakietu redystrybucyjnego Visual C...
  - ☒ Projektant klas
  - ☒ Narzędzia profilowania dla języka C++
  - ☒ Uniwersalny zestaw SDK środowiska CRT dla syst...
  - ☒ Windows 8.1 SDK
  - ☒ Obsługa języka C++/interfejsu CLI
  - ☒ Uniwersalne środowisko uruchomieniowe C syst...
  - ☒ ReadyRoll for VS2017
  - ☒ SQL Prompt Core
  - ☒ GitHub Extension for Visual Studio
  - ☒ Biblioteki i kompilatory języka Visual C++ do ob...

Spis składników instalowanych wraz z programem Visual Studio

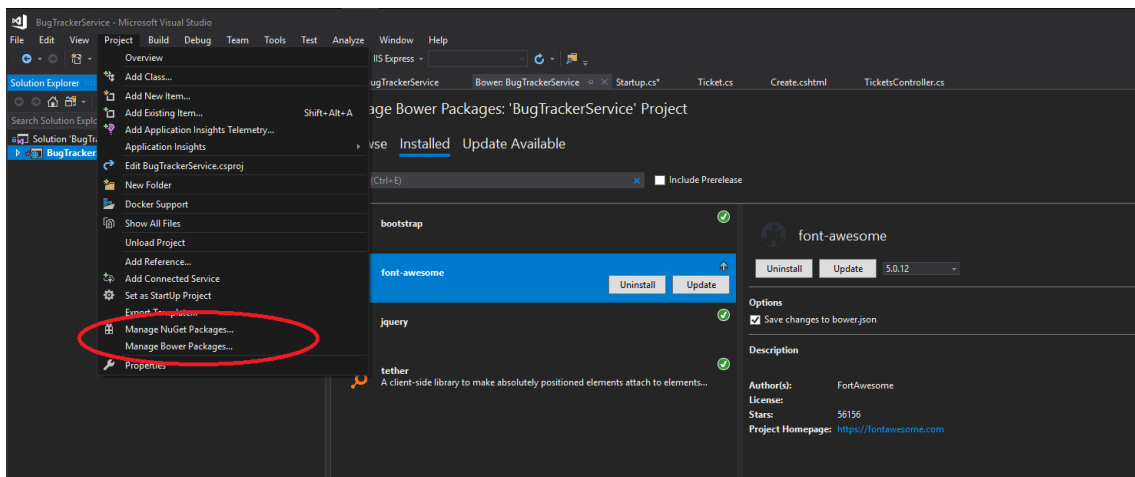
### 3.2.2. Konfiguracja

Po stworzeniu projektu, w zakładce **Project->Manage NuGet Packages** wyszukujemy pakietów: **Microsoft.AspNetCore.All**, **Microsoft.VisualStudio.Web.CodeGeneration.Design**, **Microsoft.EntityFrameworkCore.Tools** i instalujemy je w najnowszej wersji



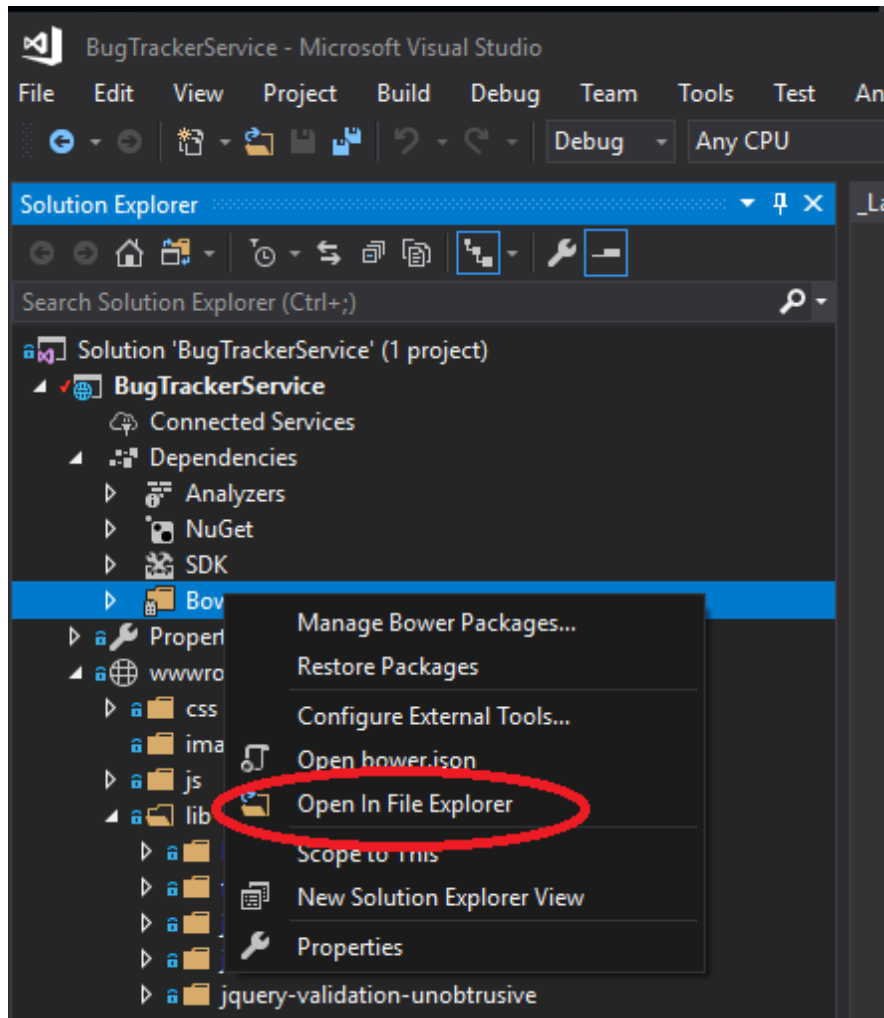
Spis pakietów potrzebnych z NuGet

Kolejnym etapem jest dodanie pakietu bootstrap4 oraz fontawesome do projektu. Wchodzimy w zakładkę **Project->Manager Bower Packages** i instalujemy bootstrap w wersji 4.x oraz font-awesome w wersji 5.x

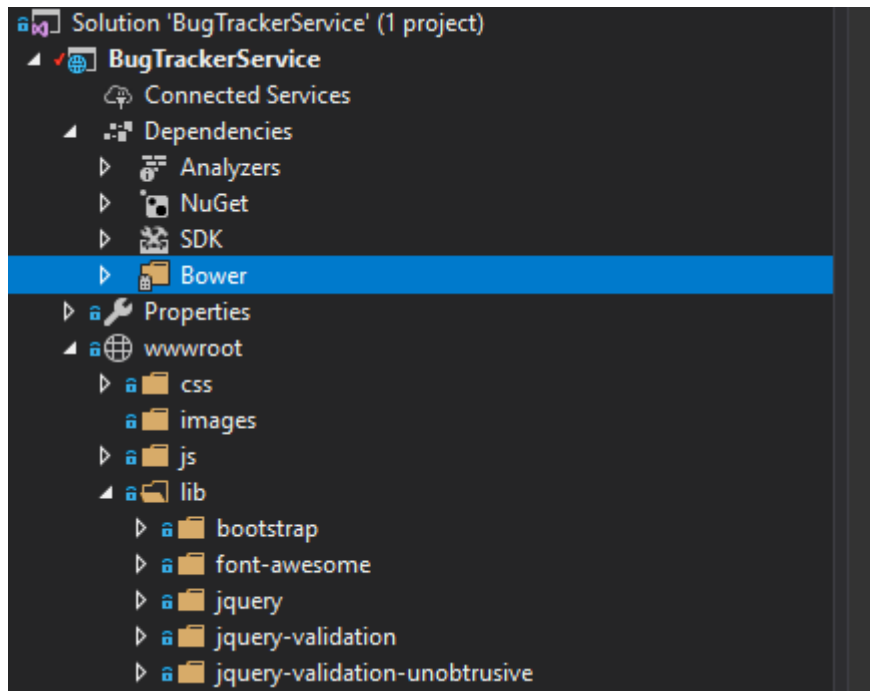


Wejście do poszczególnych instalatorów pakietów

Następnie prawym przyciskiem myszy na **Dependencies->Bower->Open in File Explorer**, przeciągamy folder bootstrap do **wwwroot->lib**, to samo robimy z folderem font-awesome. W pliku **\_Layout.cshtml** dodajemy ścieżki dostępu do **bootstrap.css** oraz **fontawesome-all.css** – jeżeli wszystko dobrze zrobiliśmy obydwie biblioteki są dostępne.



Instrukcja dodania bibliotek



Foldery w wwwroot->lib potrzebne do korzystania z bibliotek

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>@ViewData["Title"] - BugTrackerService</title>

  <environment include="Development">
    <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.css" />
    <link rel="stylesheet" href="~/css/site.css" />
    <link rel="stylesheet" href="~/lib/font-awesome/web-fonts-with-css/css/fontawesome-all.css" />
  </environment>
```

Miejsce dodania ścieżek dostępu do bibliotek

### 3.3. Opis działania kodu

### 3.4. Klasy

#### 3.4.1. Klasa Account

##### 3.4.1.1. Controller

##### 3.4.1.2. Model

##### 3.4.1.3. View

#### 3.4.2. Klasa Ticket

##### 3.4.2.1. Controller

##### 3.4.2.2. Model

##### 3.4.2.3. View

#### 3.4.3. Klasa FileUploadHelperExtensions



### 3.5. Opis działania wybranych funkcji

#### 3.5.1. Funkcja DownloadFile

### 3.6. Graficzna reprezentacja schematu bazy danych

## **4. Podsumowanie**

A tu będzie podsumowanie

## **5. Bibliografia**