

**Dokumentacja projektu**

Przedmiot: **Programowanie**

Tytuł projektu: „Książka Dyspozytora”

Prowadzący: Wykonawcy:

*Dr Marek Jaszuk Krzysztof Kostrząb W57826  
 Daniel Kolasa W57821  
 Jakub Świątek W57463*

Rzeszów, 2018

# Opis założeń projektowych.

Założeniem projektowym było wykonanie dowolnej aplikacji desktopowej. Naszym projektem jest aplikacja bazodanowa wykonana w C# i platformie .NET. Środowisko bazy danych to baza lokalna MS Access.

## Wymagania biznesowe.

Książka Dyspozytora to aplikacja dedykowana dla Dyspozytora Gazowego. Infrastruktura sieci i instalacji gazowych jest bardzo rozbudowana, dlatego też częstym zjawiskiem są awarie dotyczące tej materii. Jedną z codziennych czynności Dyspozytora jest rejestrowanie zgłoszeń, które związane są z awaryjną sytuacją na sieci gazowej. Ważnym jest by to narzędzie było przyjazne i intuicyjne w obsłudze, a związku z czym koniecznym jest stworzenie takiego interfejsu aby w sposób czytelny możliwym było rejestrowanie danego zdarzenia. Taki system może w znacznym stopniu przyspieszyć pracę dyspozytora, a zastosowanie wcześniej zdefiniowanego słownika wyboru spowoduję zminimalizowanie ewentualnie popełnianych błędów i ujednolici wpisy w bazie.

# Specyfikacja wymagań.

## Wymagania niefunkcjonalne.

2.1.1. Zadaniem aplikacji będzie archiwizacja zdarzeń na sieci gazowej, awarii, oraz działań podjętych przez Dyspozytora.

2.1.2. System musi umożliwiać zapisywanie i archiwizowanie wszystkich zdarzeń i czynności jakie mają wpływ na działanie i bezpieczeństwo sieci gazowej. System będzie służyć do przechowywania informacji m. innymi: dane kontaktowe, adresy, zakres prac, stopień awarii, ilość odbiorców w obszarze awarii. Prowadzona w ten sposób dokumentacja umożliwi ewidencjonowanie prac prowadzonych na danym gazociągu lub stacji oraz ułatwi przekazywanie obowiązków w trakcie obejmowania zmiany przez dyspozytora.

2.1.3. Aplikacja powinna pozwalać na rozbudowę o kolejne moduły (sekcje) lub zmianę konfiguracji bez wpływu na pozostałą część aplikacji.

2.1.4. Program musi zapisywać wszystkie dane w dedykowanej bazie danych – baza lokalna MS Access.

2.1.5. Dostęp do wprowadzania „rekordów” będzie posiadała osoba z odpowiednimi uprawnieniami – użytkownik o unikatowym loginie oraz haśle.

## Wymagania funkcjonalne.

Architektura aplikacji powinna być prosta, przyjazna i intuicyjna dla użytkowników aplikacji.

System powinien składać się z wyraźnie wydzielanych sekcji lub zakładek dedykowanych danej funkcjonalności:

* + - 1. Zgłoszenie
      2. Informacje o awarii
      3. Informacje o wykonanych pracach
      4. Sprawca
      5. Obsługa awarii
    1. Każda zakładka „Zgłoszenie” powinna zawierać co najmniej informacje takie jak:

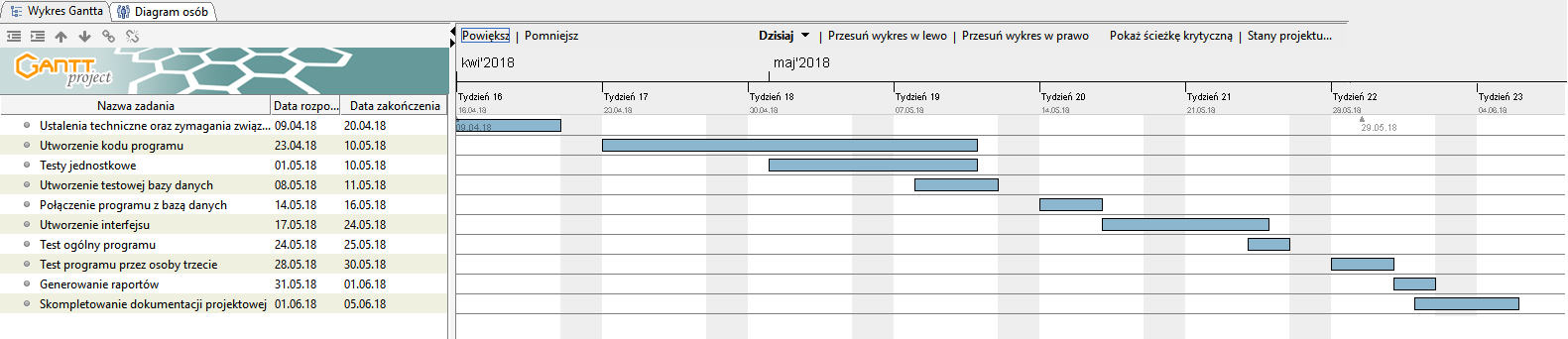
1. Imię, Nazwisko zgłaszającego
2. Forma zgłoszenia
3. Typ zgłoszenia
4. Telefon
5. Treść zgłoszenia
6. Dane adresowe zgłaszającego
7. Datę i godzinę zgłoszenia
8. Datę i godzinę planowanej realizacji
   * 1. Każda zakładka „Informacje o awarii” powinna zawierać co najmniej informacje takie jak:
9. Informacje o obiekcie i awarii
10. Dane adresowe obiektu
11. Stacje zasilającą
    * 1. Zakładka „Informacje o wykonanych pracach” powinna zawierać co najmniej informacje takie jak:
12. Typ prac
13. Przyczyna awarii
14. Osobę odpowiedzialną
15. Data i godzinę rozpoczęcia prac
16. Datę i godzinę zakończenia prac
17. Opis przyczyny awarii
18. Wyłączonych odbiorców
    * 1. Zakładka „Sprawca” ” powinna zawierać co najmniej informacje takie jak:
19. Osoba fizyczna/prawna
20. Imię, Nazwisko
21. Dane firmy
22. Ewentualne uwagi
    * 1. Zakładka „Obsługa awarii” powinna zawierać co najmniej informacje takie jak:
23. Opis wykonanych czynności
24. Ewentualne uwagi

# Diagram przypadków użycia.

???????????????????????????????????

# Diagram Gantta.

Poniżej przedstawiamy przygotowany przez nas diagram Gantta. Harmonogram z uwzględnionymi interwałami czasowymi potrzebnymi na wykonanie projektu. Można zaobserwować, iż najbardziej czasochłonnym etapem w wykonaniu projektu będzie samo wykonanie kody projektu oraz przygotowanie interfejsu projektu.

  
Rysunek . Diagram Gantta dla projektu Książka Dyspozytora.

# Opis techniczny.

Projekt został wykonany w języku programowania C#, natomiast cały interfejs za pomocą Windows Form Application. Baza danych wykorzystana w tym projekcie to utworzona przez nas baza lokalna w środowisku MS Access. Program wykorzystany do napisania i kompilacji kodu to środowisko Visual Studio 2017.

Pierwszy etap to połączenia naszego kodu z podlinkowaną bazą danych. Do wykorzystania połączenia z naszą baza danych wykorzystaliśmy listę poleceń.

polaczenie.ConnectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=E:\Wiedza\KD\_projekt\KD\_new.accdb; Persist Security Info = False;"; // polaczenie z baza danych

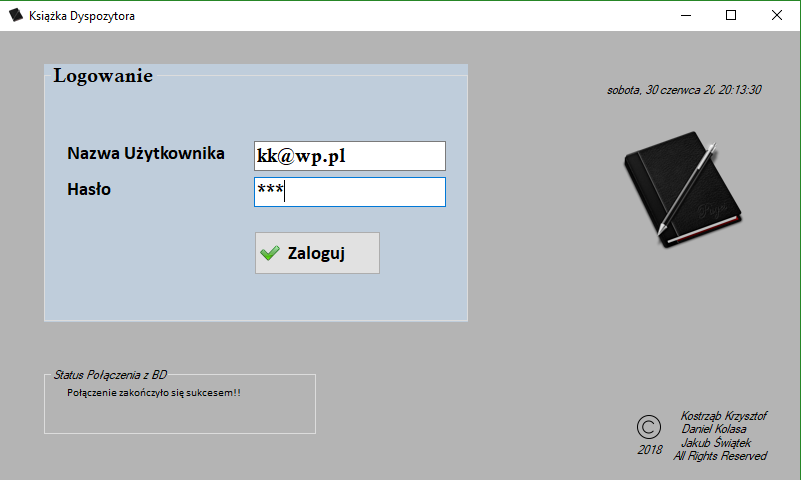
Następnie za pomocą Windows Form Application i kodu C# wykonane zostały kolejne okna aplikacji. Okna logowania, okna awarii itp.

W każdym oknie znajdują się tekst boxy wraz z odpowiednimi etykietami, dzięki czemu użytkownik posiada informacje na temat danego elemetnu.

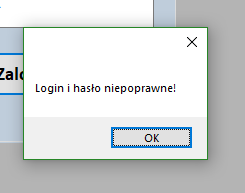
Bardzo ważnym elementem jest opcja zapisu wprowadzonych wartości do istniejącej bazy danych. Za pomocą przycisku funkcyjnego „Zapisz” i odpowiedniego kodu z języka SQL i C# wszystkie wprowadzone dane są zapisywane do Bazy Danych.

# Prezentacja warstwy użytkowej.

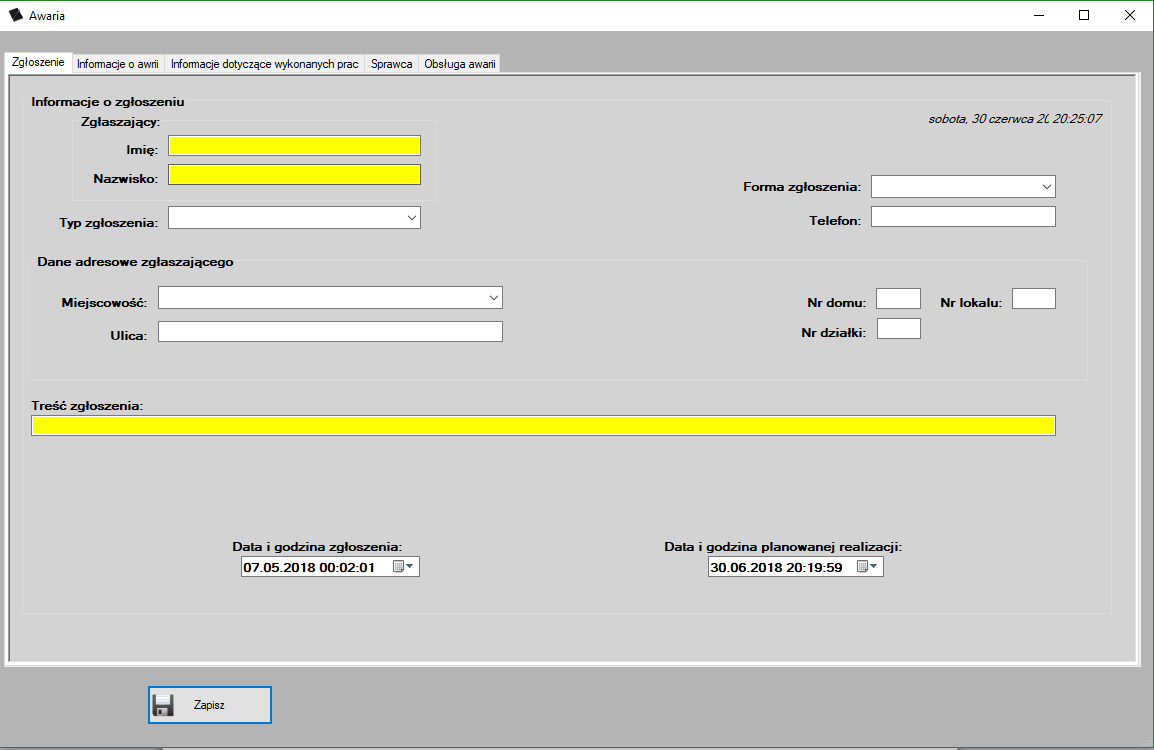
Głównym założeniem dla naszej aplikacji było to aby nasza aplikacja była prosta w obsłudze i intuicyjna. Zbędnym jest więc tworzenie rozbudowanych interfejsów z ogromną ilością opcji. Po uruchomieniu aplikacji pojawia się okno logowania do aplikacji. Użytkownik wprowadza nazwę użytkownika oraz hasło. Hasła oraz nazwy użytkowników zdefiniowane są w tej samej bazie danych. Wprowadzając login przycisk „Zaloguj” z nieaktywnego staje się aktywny. Wprowadzane hasło jest zamaskowane za pomocą symbolu „\*”. W tym oknie dialogowym znajduję się również etykieta z aktualną data i godziną, oraz status połączenia z baza danych. Poniżej przedstawiamy zrzut ekranu okna logowania.

  
Rysunek . Okno logowania do programu Książka Dyspozytora.

System sprawdza poprawność nazwy użytkownika oraz hasła – jeżeli wszystkie dane zgadzają się z bazą to pojawia się okno dialogowe informujące o poprawnym haśle. Natomiast gdy wartości są nie poprawne pojawia się informacja o błędnych danych.

  
Rysunek . Okno informujące o niepoprawnym logowaniu.

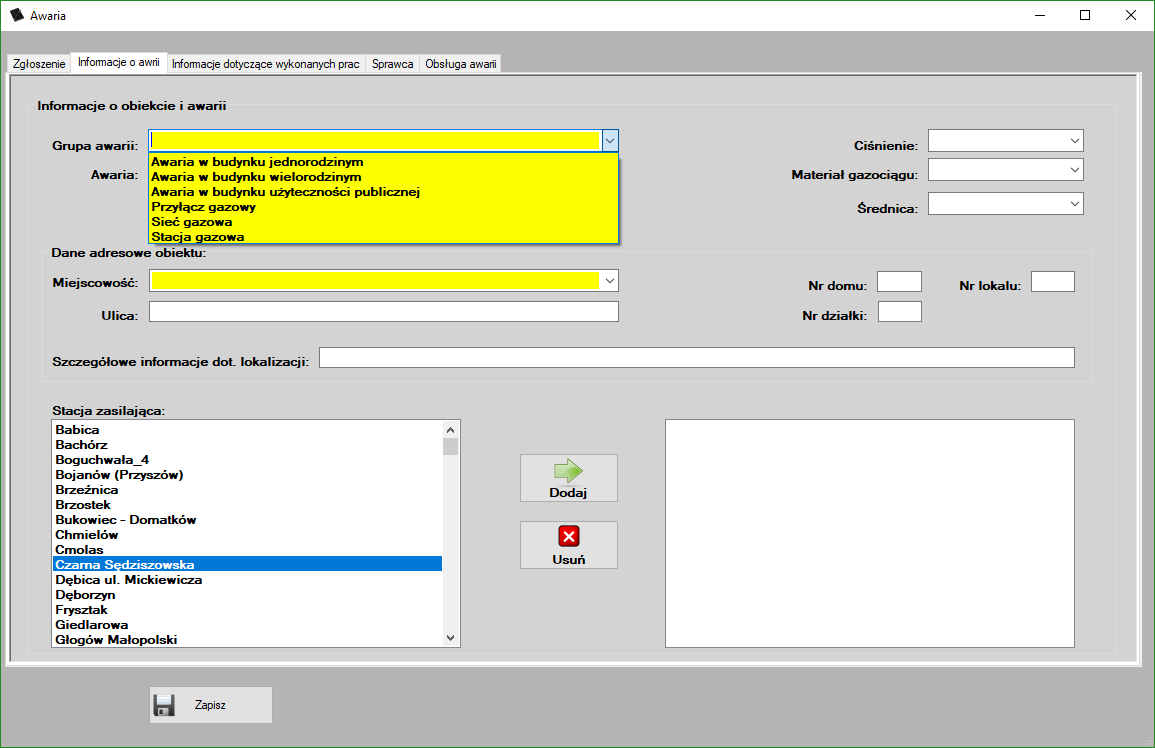
Poprawne zalogowanie do aplikacji, automatycznie uruchamia okno „Menu” Książki Dyspozytora. Przycisk „Nowa Awaria” nawiązuję połączenie z bazą danych i otwiera okno dialogowe związane z wprowadzeniem nowych wartości odnośnie Awarii. Poniżej zrzut ekranu.

Rysunek . Okno aplikacji dotyczące Awarii.

Zgodnie z wymaganiami funkcjonalnymi główne okno aplikacji zawiera zdefiniowane zakładki, oraz obszary tekstowe do wprowadzenia nowych wartości. Okna zaznaczone na żółto to wiersze które są wymagane w celu utworzenia rekordu. Po wprowadzeniu wartości okno zmienia swój status (kolor biały) informując użytkownika iż czynność została wykonana poprawnie.

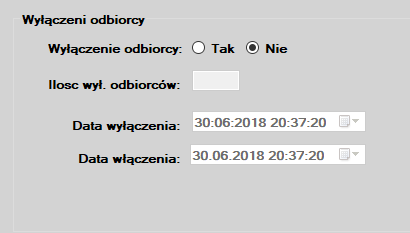
Znaczna większość okien wyboru jest powiązana z wartościami „słownikowymi”. Za pomocą list rozwijanych użytkownik w łatwy sposób może szybko wprowadzić niezbędne wartości. Wszystko to prowadzi do ujednolicenia wpisów w bazie danych wprowadzanych przez różnych użytkowników.

TextBox o etykiecie „Miejscowość” posiada zaimplementowany z bazy danych spis wszystkich miejscowości występujących w województwie podkarpackim. Ten kolejny czynnik znacznie ułatwia pracę i minimalizuję błędy w nazewnictwie. Poniżej zrzuty ekrany z wybranych okien.

  
Rysunek . Zakładka informacja o awarii.

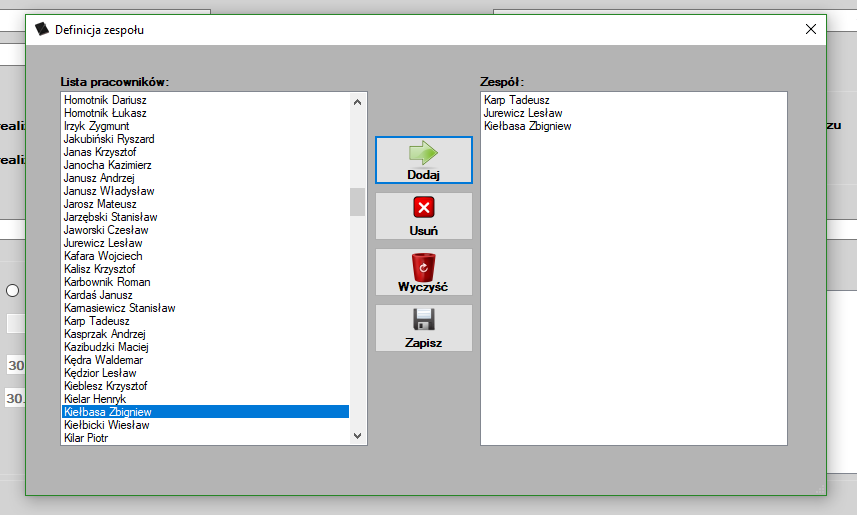
W zakładce „Informacje o awarii” utworzone zostały również dwa przyciski. Służą one do dodawania i usuwania wartości związanych z opisem stacji zasilającej. Gdy wiersz nie będzie zaznaczony wówczas pojawia się komunikat informujący o braku wyboru.

Kolejna zakładka to „Informacje o wykonanych pracach”. Znajdują się tu również listy rozwijalne dotyczące odpowiedniego textboxa. Za pomocą znacznika „TaK” w obszarze wyłączeni odbiorcy, aktywujemy opcję wprowadzenia ilości odbiorców pozbawionych gazu oraz daty wyłączenia i włączenia.

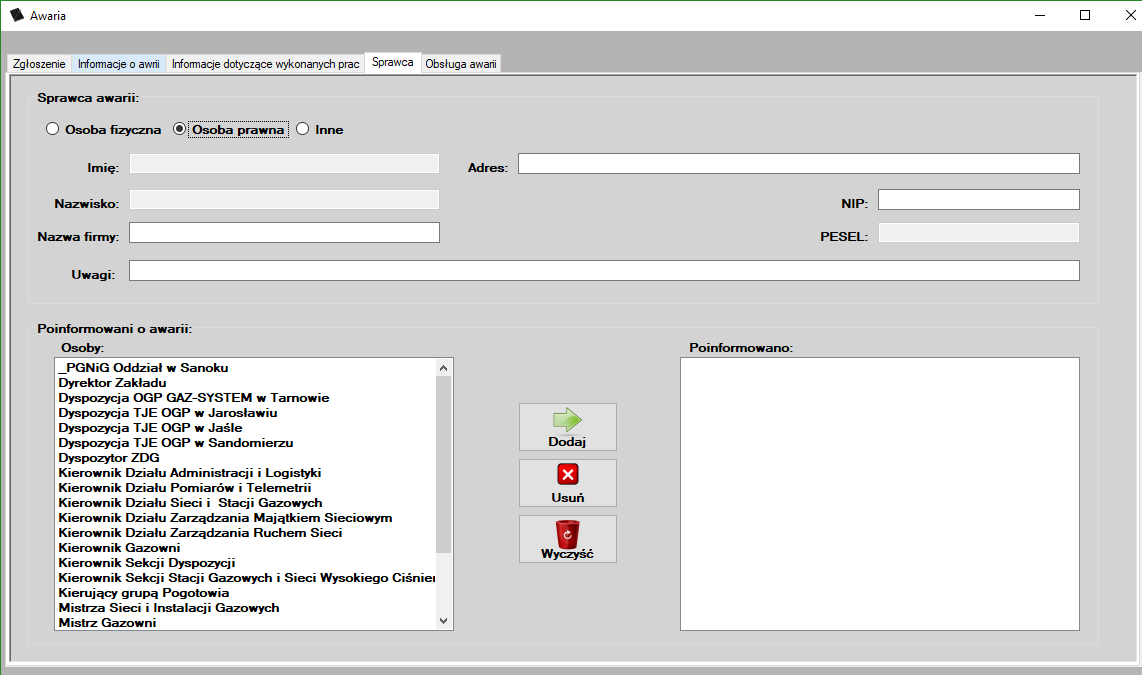


Rysunek . Okno definiujące wyłączonych odbiorców.

Przycisk „Definiuj zespół” – aktywuje okno w którym możemy przypisać skład brygady monterskiej biorącej udział przy usuwaniu danej awarii. Podobnie jak przy wyborze stacji mamy do dyspozycji przyciski służące do szybkiego przypisania pracowników.

  
Rysunek . Okno definiowania zespołu.

Okno zakładki „Sprawca” przedstawione poniżej. Zaznaczenie odpowiedniego wyboru aktywuje dostępność miejsc wpisów dotyczących sprawców awarii.

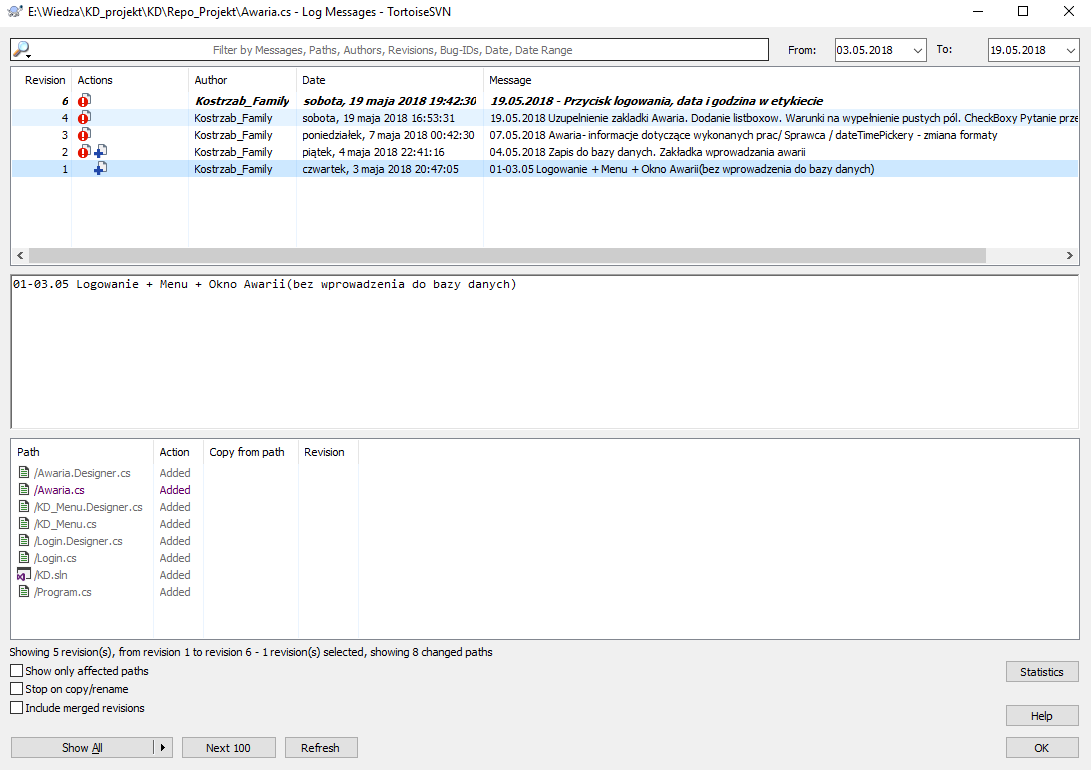
  
Rysunek . Zakładka „Sprawca awarii”.

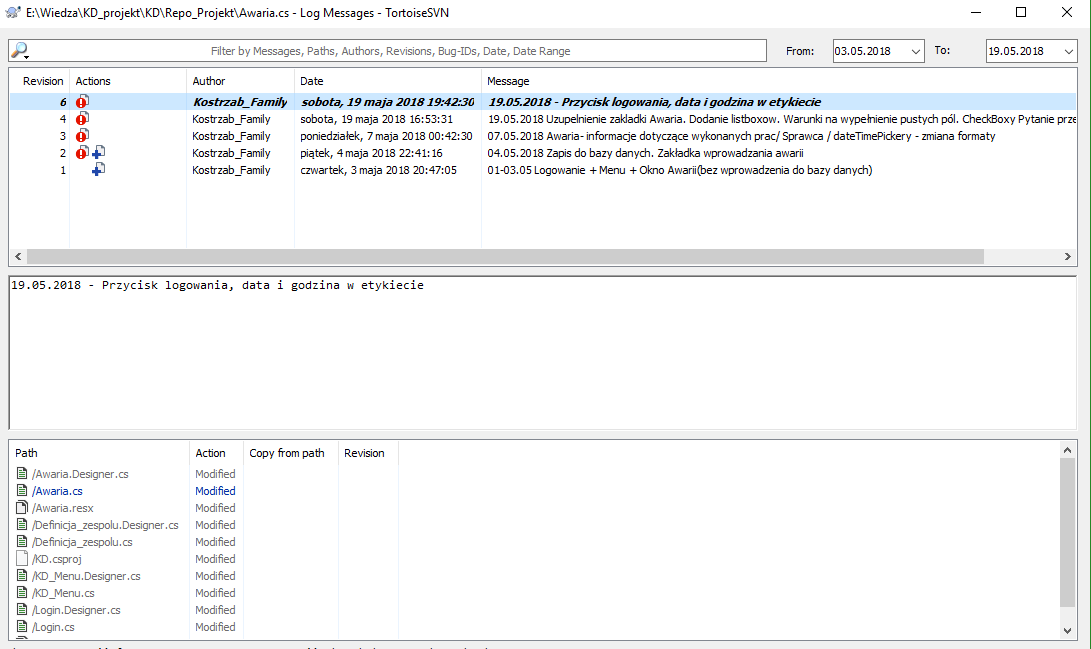
# Kontrola wersji.

Program był wykonywany w środowisku C#, kolejne wersję modyfikowanego kody były zapisywane do repozytorium lokalnego. Do tego celu wykorzystaliśmy program TortoiseSVN. Całość kodu została umieszczona na serwerze GitHub pod linkiem:

<https://github.com/Kuba203020/KD>

Poniżej przedstawiam zrzuty ekranu dokumentujące kolejne wersje Książki Dyspozytora.

  
Rysunek . Zrzut ekranu – system kontroli wersji TortoiseSVN część 1.

  
Rysunek . Zrzut ekranu – system kontroli wersji TortoiseSVN część 2.