

Sprawozdanie

Aplikacja okienkowa .NET – Task Manager

Wykonawcy: Tomasz Nowak 254021

1. Opis Zadania

Celem zadania było napisanie aplikacji okienkowej w technologii C# i WPF obsługująca przechowywanie, zapis, odczyt i prezentację danych.

Wymagania:

- Aplikacja napisana w języku programowania C#.
- Interfejs użytkownika powinien wykorzystywać technologię WPF.
- Należy zadbać o persystencję danych w aplikacji (dane nie powinny znikać po zamknięciu aplikacji, po ponownym uruchomieniu aplikacji dane powinny być dalej dostępne).
- Utworzenie bazy danych ORM i jej obsługa w technologii *Entity Framework*.
- Możliwość zapisu i odczytu danych w obrębie aplikacji oraz do i z bazy danych.
- Możliwość ręcznego wprowadzania danych do aplikacji, np. w formie formularza.
- Przeprowadzenie walidacji wprowadzanych danych i danych pobieranych z sieci.
- Graficzna prezentacja danych w postaci wykresów, tabel, list, itp.
- Wykorzystanie kolekcji obiektów do obsługi modeli danych (np. do filtrowania czy wyszukiwania odpowiednich danych w bazie).
- Obsługa kontrolek i wyjątków.
- Aplikacja powinna obsługiwać połączenie sieciowe i komunikację z zewnętrznym serwerem API.
- Wymiana danych z API powinna obsługiwać format JSON/HTML/XML lub podobny.
- Wygenerowanie dokumentacji przy użyciu wybranego generatora.

dodatkowe:

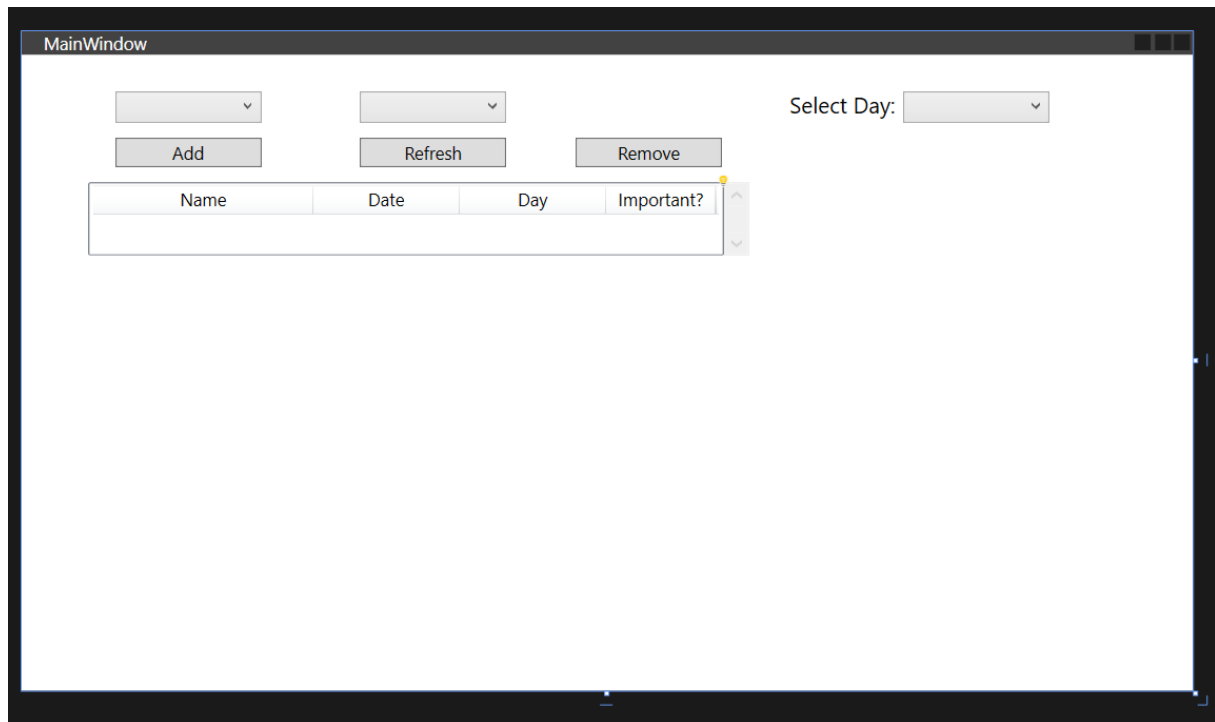
- Utworzenie testów jednostkowych.

Wybrane technologie

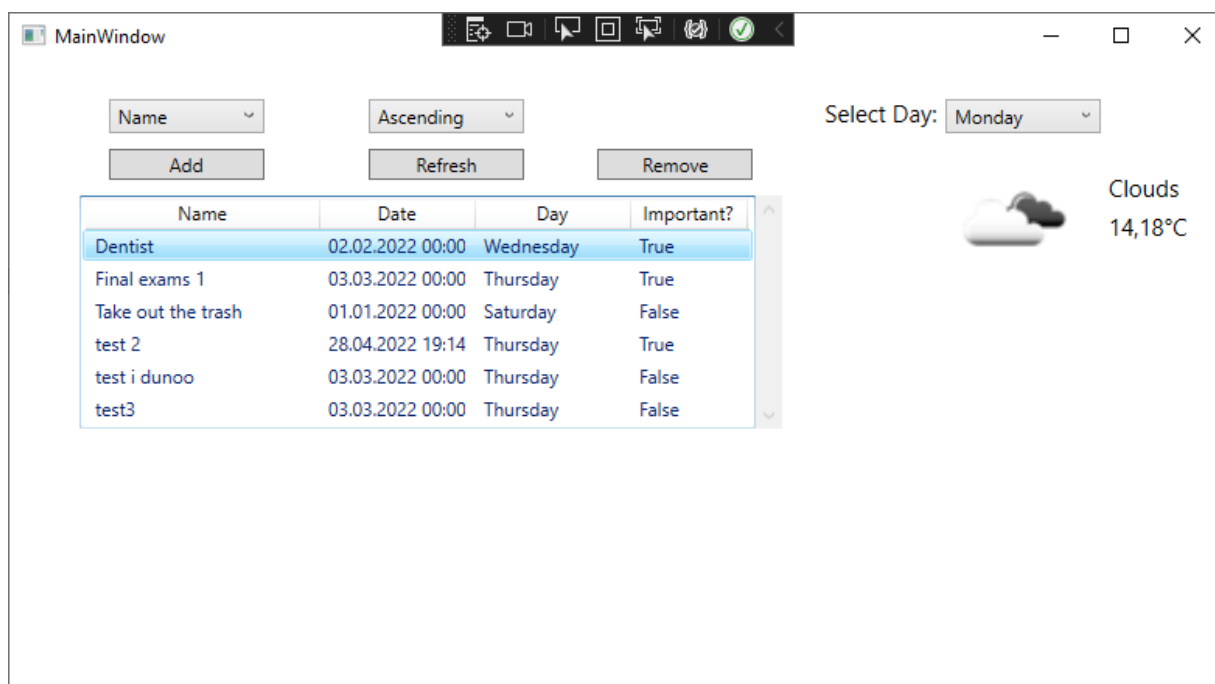
- Edytor Microsoft Visual Studio
- Biblioteka do aplikacji okienkowych WPF (Windows Presentation Foundation)
- Entity Framework – narzędzie do mapowania obiektowo-relacyjnego (ORM)
- System kontroli wersji Git (GitLab lub GitHub)
- Dokumentacja wygenerowana przy pomocy DocFX
- OpenWeatherMap – One Call API (<https://openweathermap.org/api/one-call-api>)
- Newtonsoft.Json – biblioteka do przetwarzania formatu Json

2. Program

2.1. Działanie programu

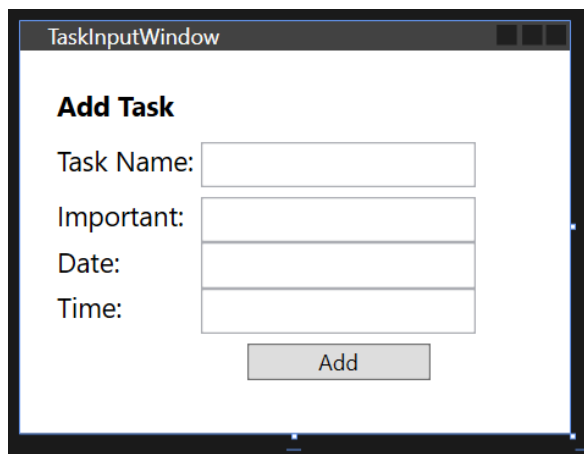


Rysunek 1 Wygląd w edytorze



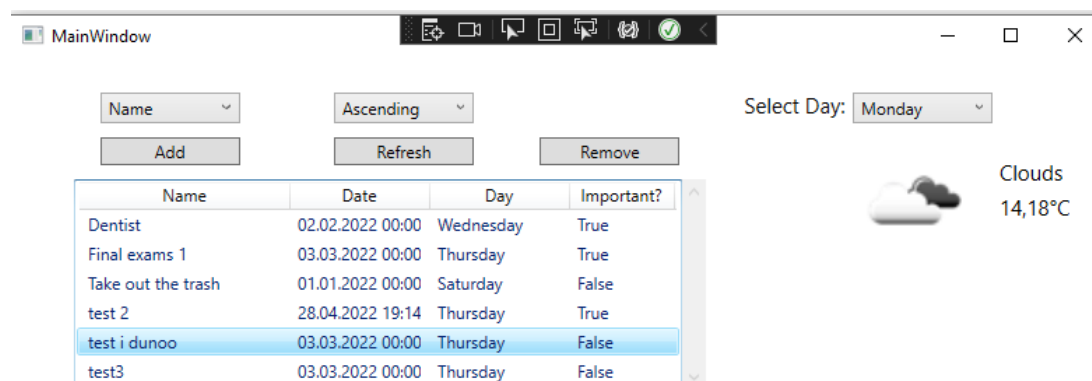
Rysunek 2 Wygląd UI po włączeniu z przykładowymi danymi

Przycisk 'Add' Wyświetla dodatkowe okienko pozwalające wpisać informacje o zadaniu, które chcemy dodać. Obsługuje ono wyjątki, więc jeśli wprowadzone dane będą błędne, to program się nie scrashuje.



The image shows a dialog box titled "TaskInputWindow". Inside, there is a section titled "Add Task". Below this title, there are four labels with corresponding input fields: "Task Name:", "Important:", "Date:", and "Time:". Each label is followed by a rectangular text input box. At the bottom center of the dialog, there is a button labeled "Add".

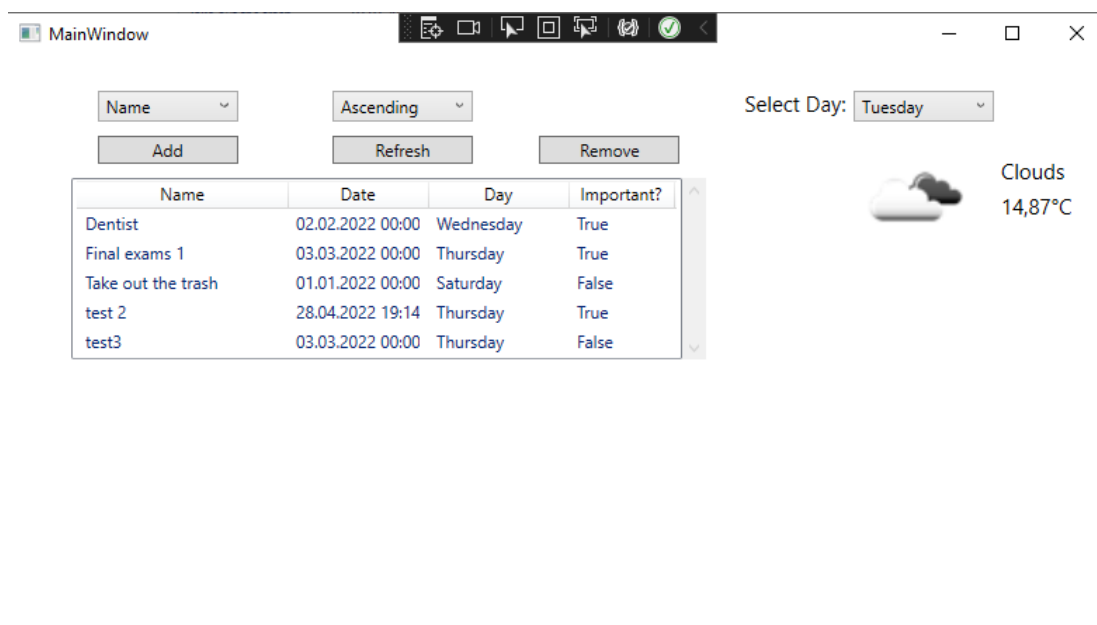
Po zaznaczeniu na liście elementu można kliknąć przycisk 'Remove' który usunie ten element z listy i bazy danych. Element znika na zawsze i nie pokaże się ponownie po zresetowaniu programu.



The image shows the main application window titled "MainWindow". It features a toolbar with icons for file operations and a status bar. Below the toolbar, there are three dropdown menus: "Name", "Ascending", and "Select Day: Monday". Below these are three buttons: "Add", "Refresh", and "Remove". The main area contains a table with the following data:

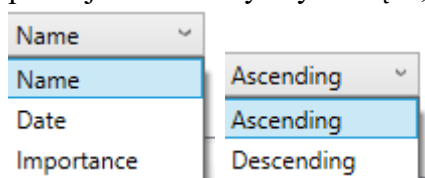
Name	Date	Day	Important?
Dentist	02.02.2022 00:00	Wednesday	True
Final exams 1	03.03.2022 00:00	Thursday	True
Take out the trash	01.01.2022 00:00	Saturday	False
test 2	28.04.2022 19:14	Thursday	True
test i dunoo	03.03.2022 00:00	Thursday	False
test3	03.03.2022 00:00	Thursday	False

To the right of the table, there is a weather widget showing a cloud icon, the text "Clouds", and the temperature "14,18°C".

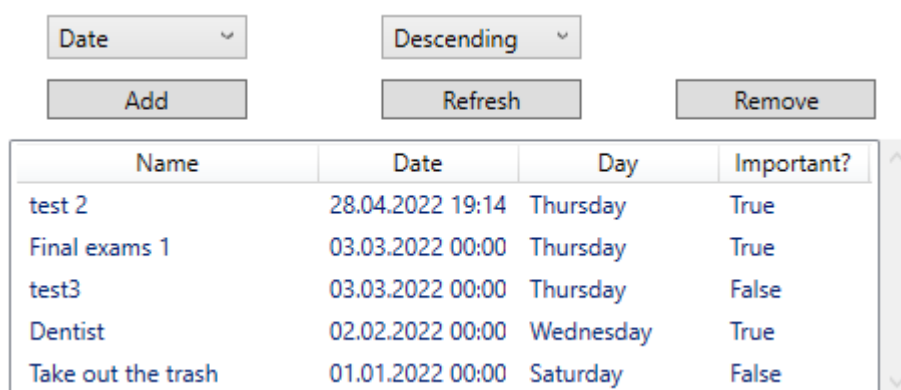


Przycisk 'Refresh' odświeża listę, jest on dodany na wypadek, gdyby na przykład po dodaniu czegoś do listy, nie wyświetliło się.

Te dwa pojemniki ponad przyciskami są ComboBoxami i służą one do sortowania listy. ComboBox z lewej strony zaznacza wartość po której chcemy sortować, natomiast w tym po prawej zaznaczamy czy rosnąco, czy malejąco.

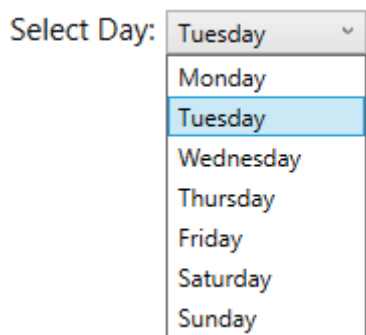


Przykład:

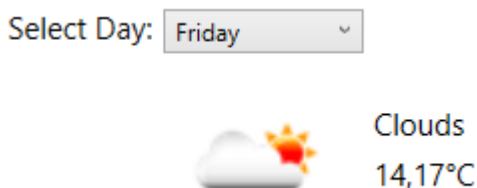


Rysunek 3 Sortowanie daty malejąco

Na prawo od listy znajduje się kolejny ComboBox, w którym zaznacza się dni tygodnia.



Zaznaczony dzień tygodnia będzie tym najbliższym od bieżącej daty. Po zaznaczeniu dnia pokażę się przewidziana temperatura i pogoda na ten dzień razem z ikonką.



Jest to zaimplementowane za pomocą API z OpenWeatherMap, nazywającego się OneCall. Za pomocą jednego zwołania do serwera jesteśmy w stanie otrzymać wszystkie pożądane dane w formacie Json. Następnie są one przetwarzane i wyświetlane. Do przetworzenia Json została użyta paczka Newtonsoft.Json dostępna do pobrania w pakietach NuGet w Microsoft Visual Studio.

Nie da się tego pokazać bez filmiku, ale dane nie usuwają się po zamknięciu aplikacji, więc po jej ponownym uruchomieniu one dalej tam będą.

3. Podsumowanie

Najtrudniejszą częścią zadania było stworzenie nie brzydkiego UI. Wstawianie nowych elementów lub zmienianie wielkości tych które już były często powodowało, że jakiś inny element/y zmieniały rozmiar, często w nieprzewidywalny sposób. Niektóre elementy też zachowują się inaczej niż by się oczekiwało.

Stworzenie takiej aplikacji w takim języku jak C++ byłoby znacznie trudniejsze. Bardzo dużo rzeczy dzieje się tutaj w tle, dzięki czemu nie trzeba się martwić o drobiazgi, co znacznie ułatwia pracę.

Microsoft Visual Studio bardzo często daje sugestie kodu, które można od razu zaimplementować klikając [TAB]. Z mojego doświadczenia, czasem kod pisze się pisząc tylko pierwszą linijkę, a potem [TAB], [ENTER], [TAB], [ENTER] itd. Bardzo zwiększa to QoL.

Dostępne są biblioteki takie jak Newtonsoft.Json, dzięki którym nie trzeba samemu pisać kodu który przetwarza Json.

Używanie baz danych w Entity Framework jest bardzo proste. Wystarczy napisać zwykłą klasę jakiegoś obiektu i kontekst, a system zajmie się resztą. Wyłuskiwanie i wstawianie danych z powrotem do tej bazy jest też bardzo łatwe i odbywa się w 1, czy 2 prostych liniijkach kodu.

Podłączenie aplikacji do internetu oraz używanie API też nie jest skomplikowane.

Podsumowując, platforma .NET jest łatwa w użytku i można robić w niej rzeczy które były by skomplikowane w językach takich jak C, czy C++ bardzo prosto.