****

**Programowanie obiektowe C#**

*MyOwn Capybara Calendar*

Prowadzący: Autor:

mgr inż. Ewa Żesławska *Adrian Klamut, Filip Kowalczyk*

nr albumu: 117805, 117808

Kierunek: INF/Ist/S/2020/21, grupa lab 2

Rzeszów 2022

Spis treści

[**1.** **Opis założeń projektu** 4](#_Toc95470213)

[**2.** **Wymagania funkcjonalne i niefunkcjonalne** 4](#_Toc95470214)

[**2.1** **Funkcjonalne** 4](#_Toc95470215)

[**2.2** **Niefunkcjonalne** 4](#_Toc95470216)

[**3.** **Diagramy UML** 4](#_Toc95470217)

[**4.** **Opis techniczny projektu** 5](#_Toc95470218)

[**5.** **Prezentacja warstwy użytkowej projektu** 6](#_Toc95470219)

[**6.** **Podsumowanie** 9](#_Toc95470220)

*.*

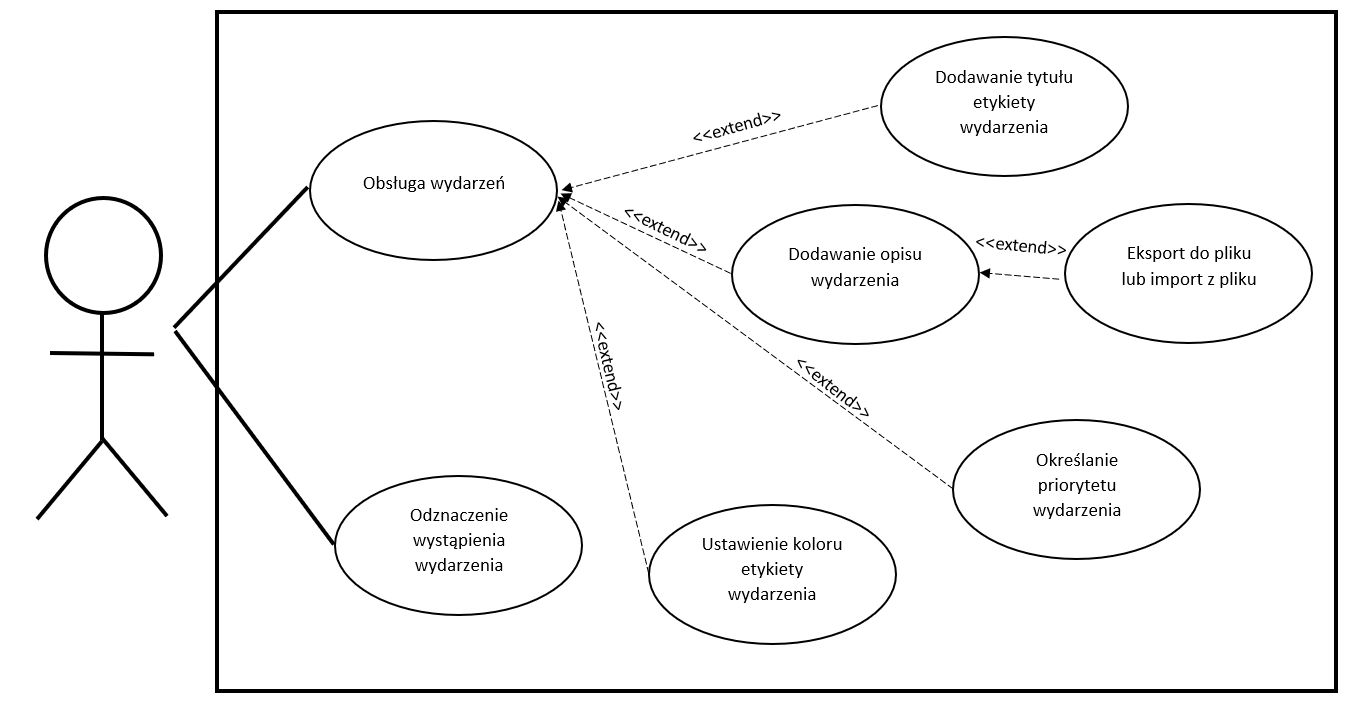
1. **Opis założeń projektu**

Utworzenie aplikacji do zarządzania kalendarzem, który posiada rozbudowaną funkcjonalność pozwalającą tworzyć dedykowane dla każdego dnia notatki, oraz specjalne tagi których zadaniem jest ustawienie etykiety dla naszej kartki z kalendarza.

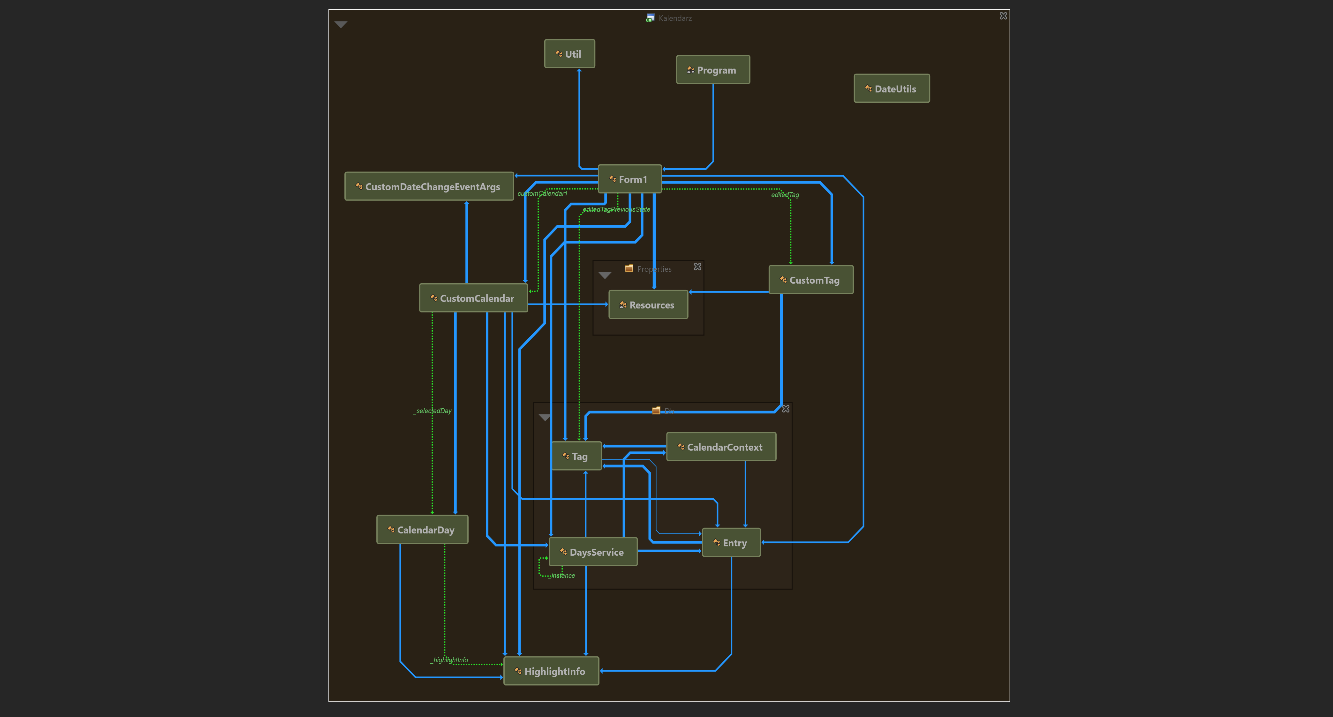
1. **Wymagania funkcjonalne i niefunkcjonalne**
   1. **Funkcjonalne**
   * Możliwość wybrania daty z kalendarza
   * Możliwość utworzenia notatki dla wybranego przez użytkownika dnia.
   * Edycja oraz usunięcie utworzonej notatki.
   * Możliwość utworzenia etykiety dla wybranej daty z kalendarza.
   * Edycja oraz usunięcie utworzonej etykiety.
   * Eksport i import notatki z/do pliku tekstowego na dysku lokalnym.
   * Można wprowadzić wiele etykiet dla danej daty.
   1. **Niefunkcjonalne**

* Wybieranie dni, dodawanie i edycja notatek oraz zarządzanie etykietami dokonywane jest przez użytkownika.

1. **Diagramy UML**
   1. Diagram przypadków użycia



* 1. Diagram klas



1. **Opis techniczny projektu**

* Wykorzystany język C#, framework .NET, Windows Forms
* Środowisko programistyczne C#: MS Visual Studio 2022
* Wymagany jest komputer z systemem operacyjnym Windows 7 lub nowszym
* Wymagane jest połączenie z Internetem lub lokalna baza danych
* Wymagane jest przynajmniej 2GB pamięci RAM i procesor Intel Pentium lub nowszy
* Wymaganie jest 10MB wolnego miejsca na dysku
* Dane są przechowywane w dwóch tabelach w bazie danych (u nas MariaDB)
* Diagram ERD wygląda następująco:

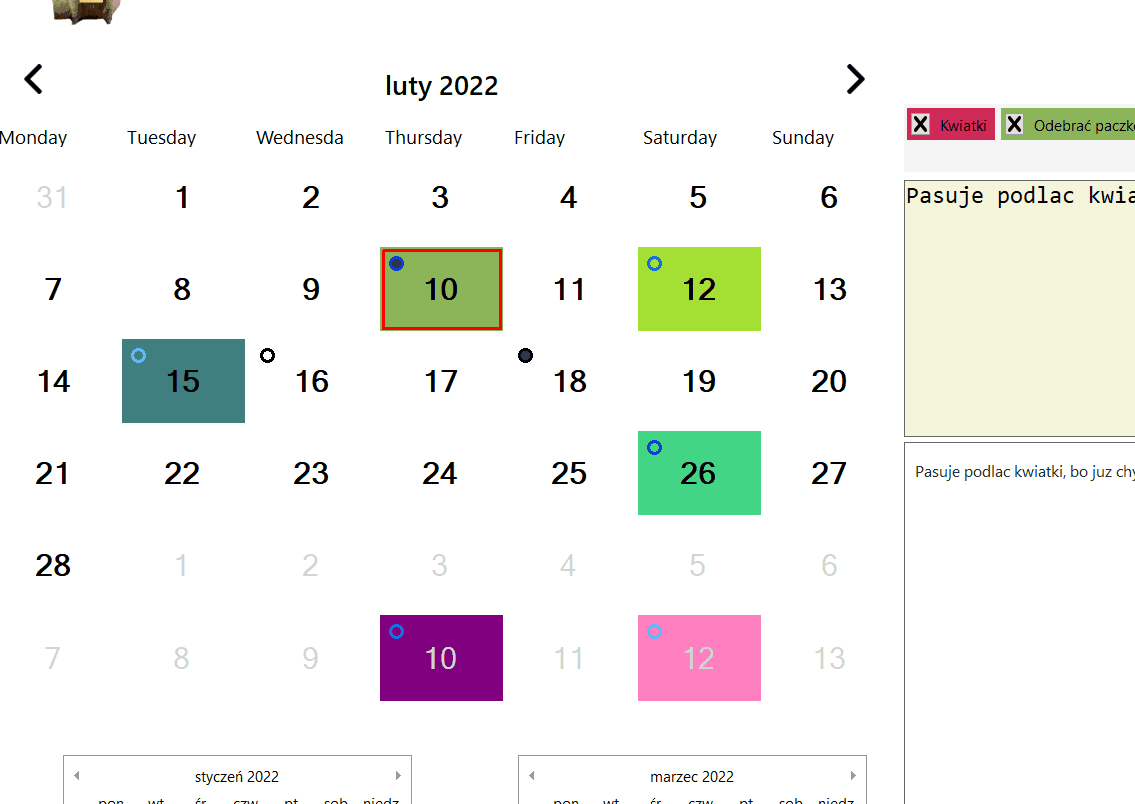
Diagram ERD


* Połączenie z bazą danych i edycja danych realizowane są przy użyciu ORM - Entity Framework Core

1. **Prezentacja warstwy użytkowej projektu**

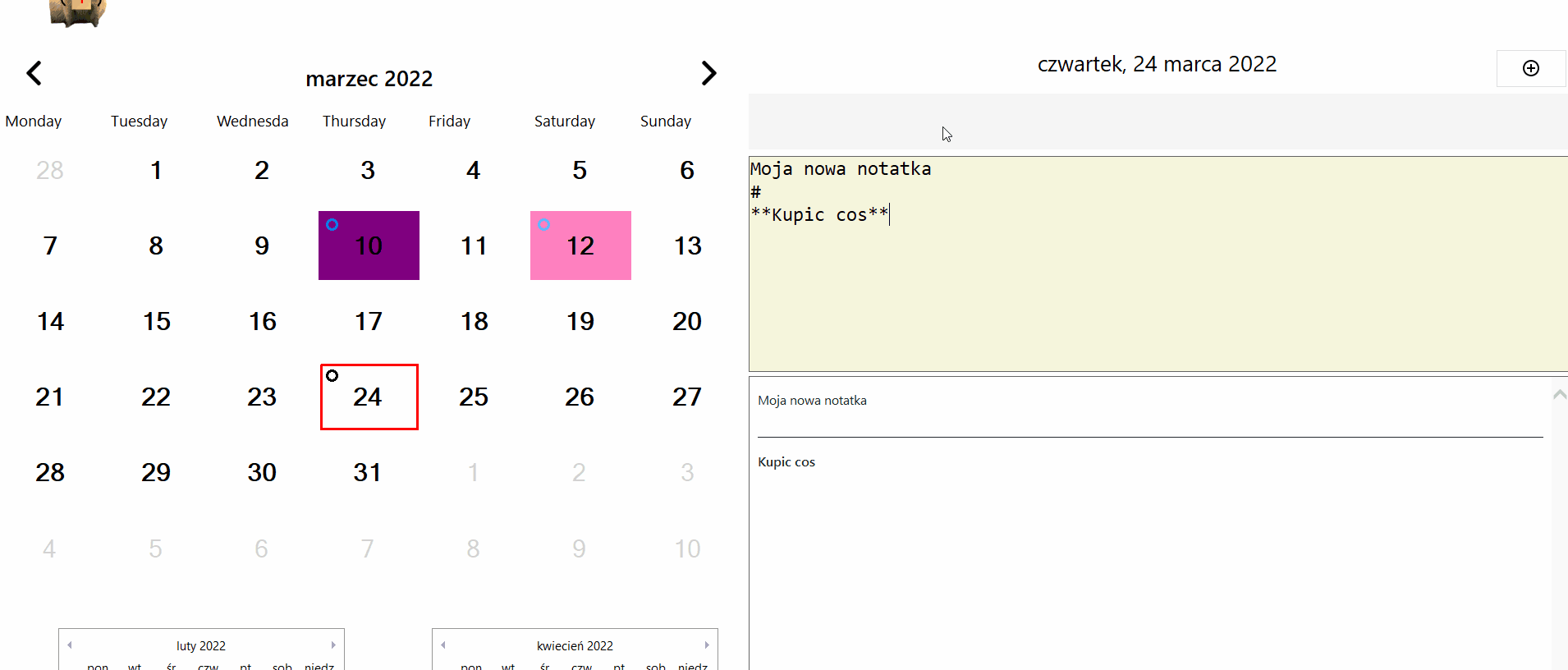
5.1 Wybranie daty:

- Kliknięcie w odpowiedni dzień wyznacza nam edytowaną datę.



5.2.Dodawanie notatki oraz edycja

- Po wybraniu dnia dla którego chcemy utworzyć wpis, po prawej stronie otrzymujemy pole w którym możemy utworzyć notatkę lub etykietę. Zapisywana przez nas notatka jest od razu wyświetlana poniżej w formacie .md, daje nam to dużo możliwości jeżeli chodzi o kompozycje oraz edycje naszego tekstu.

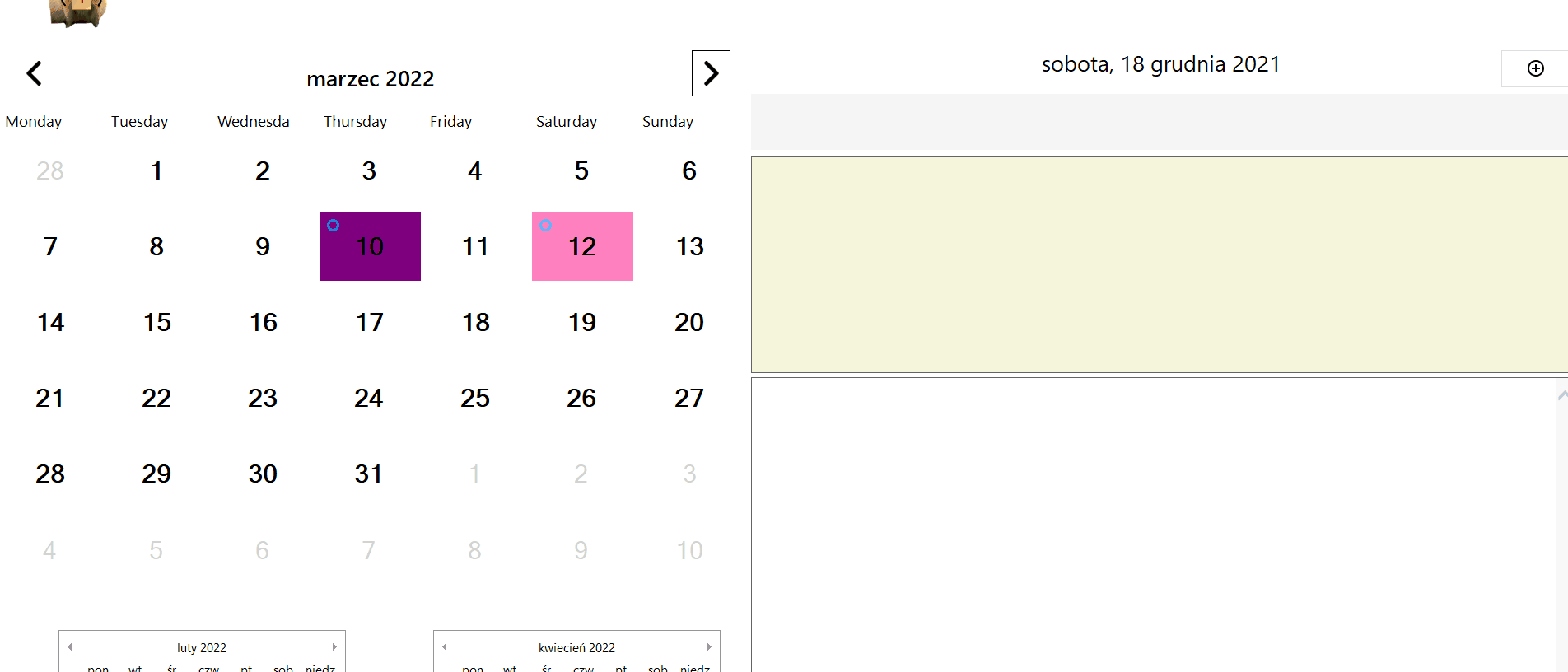


5.3.Dodawanie etykiety

- Opcja ta pozwala na dodanie etykiety dla danego dnia.

Przypisujemy naszej etykiecie nazwę, kolor, priorytet oraz własność „użyj jako tła”.

Jeżeli ustawimy jej priorytet oraz zaznaczymy „użyj jako tła”, nasza kartka z kalendarza odziedziczy kolor bo tej etykiecie która ma najwyższy priorytet



5.4.Zapis do pliku

- Poprzez kliknięcie „Eksportuj do pliku” możemy zapisać wybraną notatkę w pliku tekstowym na dysku lokalnym.

Obraz zawierający tekst

Opis wygenerowany automatycznie

5.5.Odczyt z pliku

- Poprzez kliknięcie „Otwórz z pliku” możemy przypisać do naszej notatki zawartość pliku tekstowego z dysku lokalnego.

Obraz zawierający tekst

Opis wygenerowany automatycznie

5.6.Dodatkowe funkcjonalności :

- Kopiuj tekst

- Wklej tekst

- Doklej tekst

- Skocz do dzisiejszego dnia – przenosi nas do dzisiejszej daty.

- Kliknięcie w okrągły znacznik przy kartkę z kalendarza, oznaczy to wydarzenie jako wykonane.

1. **Podsumowanie**

Projekt został wykonany zgodnie z założeniami. Pliki źródłowe znajdują się w repozytorium git pod adresem: <https://github.com/ADIOP55550/Projekt-CS>