|  |  |
| --- | --- |
| **A picture containing text, clipart, businesscard  Description automatically generated** | **Text  Description automatically generated with low confidence** |

Mateusz Pawlikowski

nr albumu: 44452

kierunek studiów: Informatyka

specjalność: Inżynieria oprogramowania

forma studiów*: studia stacjonarne*

Projekt i implementacja systemu komputerowego do wspomagania obserwacji astronomicznych

Astronomical observation support system – project and implementation

praca dyplomowa inżynierska

napisana pod kierunkiem:

dr hab. inż. Wojciech Maleika

Katedra Systemów Multimedialnych

Data wydania tematu pracy: 31.1.2022

Data dopuszczenia pracy do egzaminu: …………………………………………………………………………………………………….  
(uzupełnia pisemnie Dziekanat)

Szczecin, 2022

Spis treści

[Wstęp 3](#_Toc92387844)

[Cel pracy 3](#_Toc92387845)

[Zakres pracy 4](#_Toc92387846)

[1. Analiza istniejących rozwiązań na rynku 4](#_Toc92387847)

[1.1 System SkyMap 4](#_Toc92387848)

[1.2 System AstroViewer 5](#_Toc92387849)

[1.3 Wnioski 6](#_Toc92387850)

[2. Analiza wykorzystanych narzędzi 6](#_Toc92387851)

[2.1 .NET Core Framework 6](#_Toc92387852)

[2.2 Zintegrowane środowisko programistyczne – Visual Studio 7](#_Toc92387853)

[2.3 Języki programowania 7](#_Toc92387854)

[2.3.1 C# 7](#_Toc92387855)

[2.3.2 F# 8](#_Toc92387856)

[2.4 Windows Presentation Foundation 9](#_Toc92387857)

[2.4.1 Język opisu interfejsu użytkownika w WPF - XAML 9](#_Toc92387858)

[2.4.2 Silnik graficzny WPF 10](#_Toc92387859)

[2.4.3 Wieloplatformowość 10](#_Toc92387860)

[2.5 SQLite i DB Browser 11](#_Toc92387861)

[3. Podstawy teoretyczne zagadnień astronomicznych 11](#_Toc92387862)

[3.1 Czas w astronomii 11](#_Toc92387863)

[3.1.1 Dni juliańskie 11](#_Toc92387864)

[3.1.2 Lokalny czas gwiazdowy 11](#_Toc92387865)

[3.2 Układy współrzędnych astronomicznych 12](#_Toc92387866)

[3.2.1 Współrzędne horyzontalne 12](#_Toc92387867)

[3.2.2 Współrzędne równikowe 13](#_Toc92387868)

[3.2.3 Inne współrzędne 14](#_Toc92387869)

[3.3 Konwersje między współrzędnymi 15](#_Toc92387870)

[3.3.1 Konwersja współrzędnych równikowych do horyzontalnych 17](#_Toc92387871)

[3.3.2 Konwersja współrzędnych horyzontalnych do równikowych 17](#_Toc92387872)

[3.3.3 Konwersje pozostałych współrzędnych 17](#_Toc92387873)

[3.4 Właściwości obiektów astronomicznych 17](#_Toc92387874)

[3.4.1 Odległość od Słońca 18](#_Toc92387875)

[3.4.2 Wielkość gwiazdowa 18](#_Toc92387876)

[3.4.3 Typ widmowy 19](#_Toc92387877)

[4. Projekt systemu 19](#_Toc92387878)

[4.1 Analiza funkcjonalności 19](#_Toc92387879)

[4.1.1 Przechowywanie i pobieranie obiektów z bazy danych 19](#_Toc92387880)

[4.1.2 Wyświetlanie obiektów na mapie nieba 20](#_Toc92387881)

[4.1.3 Interaktywność i sterowanie mapy 20](#_Toc92387882)

[4.1.4 Wyświetlanie informacji o danym obiekcie 20](#_Toc92387883)

[4.1.5 Ustawienie lokalizacji 20](#_Toc92387884)

[4.1.6 Ustawienie czasu 20](#_Toc92387885)

[4.1.7 Upływ i sterowanie czasem 20](#_Toc92387886)

[4.1.8 Aktualizacja mapy 21](#_Toc92387887)

[4.1.9 Ocena warunków pogodowych 21](#_Toc92387888)

[4.2 Architektura 22](#_Toc92387889)

[4.2.1 Przegląd wzorców architektonicznych 23](#_Toc92387890)

[4.2.2 Wybór architektury 23](#_Toc92387891)

[4.2.3 Projekt architektury systemu 24](#_Toc92387892)

[4.3 Baza danych 30](#_Toc92387893)

[4.3.1 Źródło danych 30](#_Toc92387894)

[4.3.2 Projekt bazy danych 31](#_Toc92387895)

[4.4 Implementacja 32](#_Toc92387896)

[4.4.1 Algorytmy astronomiczne 32](#_Toc92387897)

[4.4.2 Obliczanie pozycji obiektów na mapie 35](#_Toc92387898)

[4.4.3 Wczytywanie obiektów astronomicznych 36](#_Toc92387899)

[4.5 Projekt interfejsu 36](#_Toc92387900)

[4.5.1 Mapa sfery niebieskiej 37](#_Toc92387901)

[4.5.2 Obiekty astronomiczne 38](#_Toc92387902)

[4.5.3 Pasek narzędzi 40](#_Toc92387903)

[4.5.4 Panel oceny warunków obserwacyjnych 41](#_Toc92387904)

[5. Testy systemu 41](#_Toc92387905)

[5.1 Weryfikacja pobierania i wyświetlania obiektów na mapie 42](#_Toc92387906)

[5.2 Weryfikacja interaktywności i sterowalności mapy 42](#_Toc92387907)

[5.3 Weryfikacja wyświetlania informacji o danym obiekcie 43](#_Toc92387908)

[5.4 Weryfikacja ustawiania lokalizacji i czasu obserwacji 43](#_Toc92387909)

[5.5 Weryfikacja upływu i sterowania czasem 44](#_Toc92387910)

[5.6 Weryfikacja oceny warunków pogodowych 45](#_Toc92387911)

[5.7 Podsumowanie testowania 45](#_Toc92387912)

[6. Wnioski i podsumowanie 45](#_Toc92387913)

[**Bibliografia** 47](#_Toc92387914)

# Wstęp

Oprogramowania komputerowe od ostatnich dziesięcioleci wspierają oraz automatyzują naszą pracę we wszelkich możliwych dziedzinach abyśmy mogli szybciej i wydajniej uzyskiwać interesujące nas dane. W przypadku takiej dziedziny jak astronomia oprogramowania komputerowe przyczyniły się do szybkiego rozwoju tej gałęzi nauki oraz ułatwiły prowadzenie obserwacji kosmosu dla osób zajmujących się tym amatorsko. Obserwacje astronomiczne nie są łatwe gdyż dużo czasu trzeba poświęcić nie tylko na samo przygotowanie sprzętu obserwacyjnego ale też na samo znalezienie interesujących nas obiektów astronomicznych. Ze względu na wyżej wymienione fakty powstał więc pomysł na utworzenie oprogramowania komputerowego wspierającego tę aktywność aby ludzie korzystający z niego mogli więcej czasu poświęcić na same obserwacje.

## Cel pracy

Celem pracy jest zaprojektowanie i implementacja oprogramowania ułatwiającego prowadzenie obserwacji astronomicznych. Użytkownik będzie mógł wprowadzić współrzędne geograficzne oraz datę i czas obserwacji dzięki czemu wygenerowana zostanie interaktywna mapa nieba w miejscu obserwacji. Dodatkowo użytkownik będzie mógł wprowadzić dane o pogodzie w swoim regionie aby określić czy warunki pogodowe nadają się do celów obserwacyjnych.

## Zakres pracy

Praca składa się z pięciu rozdziałów. Pierwszy rozdział będzie o podobnych już istniejących systemach wspierających obserwacje astronomiczne na rynku. W drugim rozdziale omówione będą narzędzia wykorzystane do utworzenia systemu wspomagania obserwacji astronomicznych. W trzecim rozdziale wyjaśnione zostaną podstawowe zagadnienia astronomiczne, które potrzebne są do utworzenia takiego systemu. W czwartym rozdziale przedstawione będzie projekt systemu czyli w jakiej architekturze został zaprojektowany, gdzie przechowywane są dane oraz w jaki sposób algorytmy astronomiczne zostały zaimplementowane. Piąty rozdział poświęcony będzie testom manualnym utworzonego systemu w celu weryfikacji czy wszystkie wymagania funkcjonalne zostały spełnione i czy system działa prawidłowo.

# 1. Analiza istniejących rozwiązań na rynku

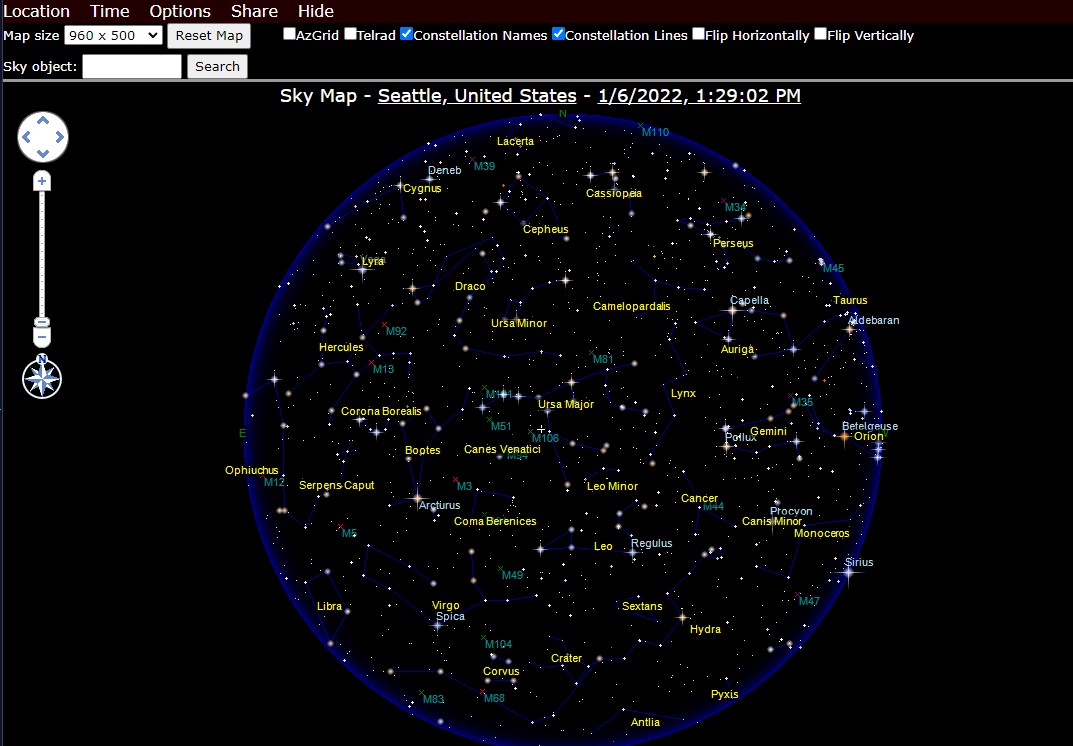
W tym rozdziale przedstawione zostaną istniejące rozwiązania systemów wspierających obserwacje astronomiczne. Będą to systemy darmowe, do których można uzyskać dostęp za pomocą Internetu. Te systemy to [SkyMap](https://www.skymaponline.net) oraz [AstroViewer](https://www.astroviewer.net/av/en/).

## System SkyMap

System SkyMap jest aplikacją webową udostępnioną pod adresem <https://www.skymaponline.net>. SkyMap utworzony został przez programistę i pasjonata astronomii o imieniu Ciprian Boboc, który rozpoczął pracę nad nim oraz wydał jego pierwszą wersję w 2010 roku.

System ten składa się z dwuwymiarowej, okrągłej mapy sfery niebieskiej, na której wyświetlane są różnego rodzaju obiekty astronomiczne takie jak gwiazdy, obiekty głębokiego nieba oraz gwiazdozbiory. W systemie można wprowadzić współrzędne geograficzne oraz datę i czas obserwacji co spowoduje aktualizację mapy aby wyświetlała obiekty widoczne dla wprowadzonych wspomnianych danych. SkyMap jest aplikacją webową i z tego względu ma ona bezpośrednie połączenie z Internetem, dzięki temu może on sam wczytać geograficzną lokalizację użytkownika po adresie IP. W systemie tym można również wpisać nazwę interesującego nas obiektu, który po wyszukaniu zostanie umieszczony w centrum mapy.

Mapa w SkyMap jest interaktywna ale również nieprzyjemna w użyciu ponieważ przybliżanie oraz obracanie mapy powoduje, że mapa znika na pewien czas aż wczyta odpowiednie obiekty. W pewnych momentach wręcz przy zmianie jakichkolwiek danych aktualizujących mapę system przestaje odpowiadać i nie dostajemy żadnych informacji. Z tego powodu system, mimo że posiada wiele przydatnych funkcjonalności, nie działa na tyle dobrze dla użytkownika końcowego, który mógłby wykorzystać go w celach obserwacyjnych.



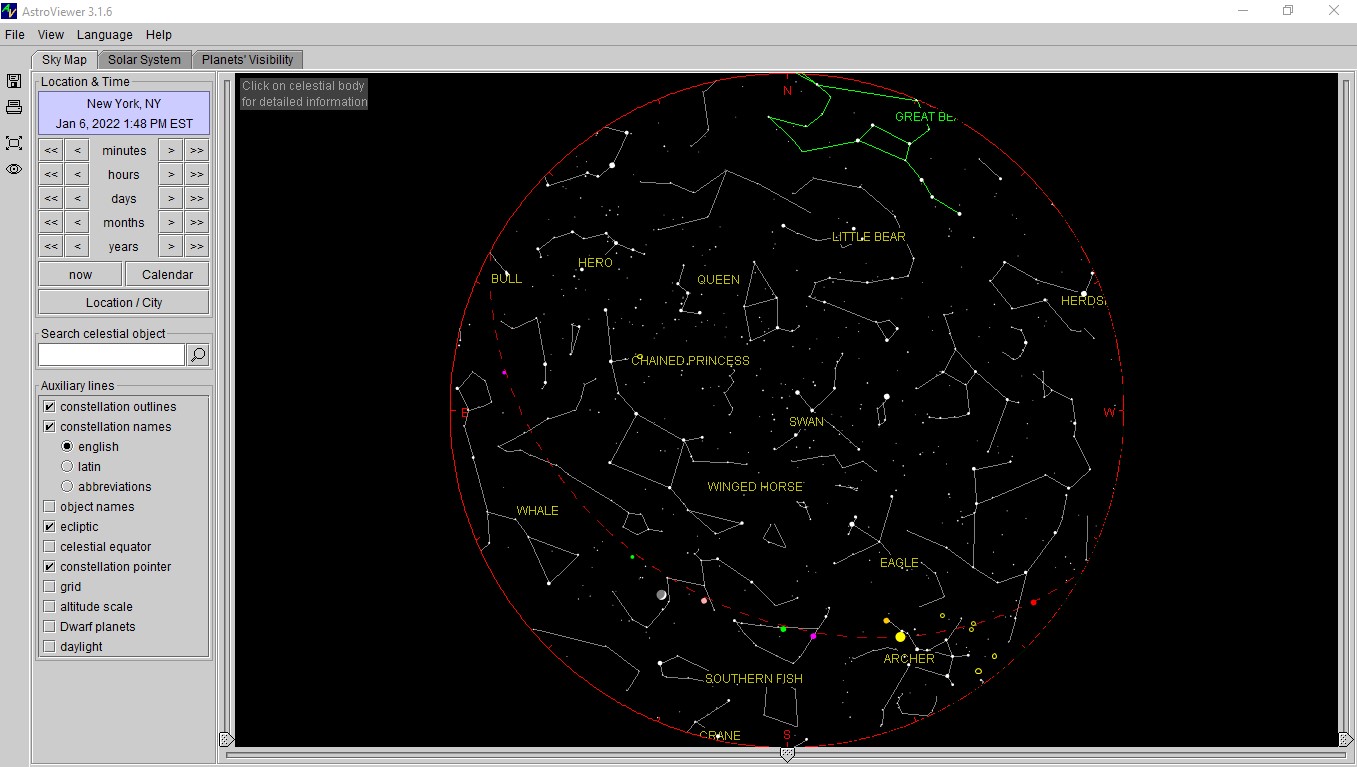
Rysunek 1 Widok aplikacji webowej SkyMap.

## System AstroViewer

AstroViewer jest interaktywną mapą nieba, która została utworzona jako aplikacja desktopowa. Ostatnia jej wersja została wydana w wersji 3.1.6 i jest dostępna za darmo do pobrania na stronie internetowej <https://www.astroviewer.net>.

System AstroViewer zawiera podstawowe funkcje, które zostały wymienione w poprzednim rozdziale poświęconym systemu SkyMap, czyli wyświetlanie mapy dla danej lokalizacji i czasu z obiektami różnego rodzaju, wyszukiwanie tych obiektów na mapie oraz wyświetlanie informacji na ich temat. W systemie istnieje dodatkowo trójwymiarowa mapa Układu Słonecznego oraz zestawienie widoczności planet dla danej lokalizacji. Aplikacja jest również stabilna i aktualizacja mapy przebiega natychmiastowo.

Podobnie jak w przypadku systemu SkyMap interaktywność mapy nie jest dobrze zrobiona dla użytkownika końcowego. W tym przypadku nie da się myszką komputera sterować mapą co jest bardzo nieintuicyjne, za to należy przesuwać odpowiednie suwaki rozmieszczone w interfejsie użytkownika.



Rysunek 2 Główny widok systemu AstroViewer.

## Wnioski

Na rynku nie brakuje systemów, które wspierają obserwacje astronomiczne. Wyżej wymienione systemy dostarczają odpowiednie dane jednak brakuje im dobrze zaprojektowanego interfejsu i w przyjemny sposób zaimplementowanej interaktywności mapy, która jest głównym i najważniejszym elementem całego systemu.

# 2. Analiza wykorzystanych narzędzi

System wspomagania obserwacji astronomicznych jest utworzony dla systemów z serii Windows firmy Microsoft, z tego powodu do utworzenia go wykorzystano popularne narzędzia pozwalające na pracę w wyżej wymienionym systemie operacyjnym.

## .NET Core Framework

Oprogramowanie tematu zostało zaprojektowane z wykorzystaniem frameworka **.NET Core w wersji 5.0**, które jest wolnym i otwartym oprogramowaniem przeznaczonym do tworzenia aplikacji w systemach operacyjnych Windows, Linux oraz macOS. Framework ten wydany został jako otwarto źródłowa implementacja platformy .NET Framework co pozwoliło na szybszy rozwój oraz na polepszenie wydajności tego środowiska. .NET Core udostępnia wiele zaimplementowanych bibliotek ułatwiających szybką pracę nad tworzeniem wydajnego oprogramowania.

## 2.2 Zintegrowane środowisko programistyczne – Visual Studio

Do szybkiego i sprawnego tworzenia oprogramowania istnieje możliwość skorzystania z tzw. zintegrowanego środowiska programistycznego (ang. integrated development environment – IDE), który dostarcza podstawowe programistyczne narzędzia takie jak edytor kodu, budowanie projektu oraz narzędzia pozwalające na debugowanie oraz testowanie go.

Jednym z takich środowisk jest Visual Studio firmy Microsoft. Dostarcza on wszystkie wyżej wymienione narzędzia oraz w pełni wspiera framework .NET. Visual Studio jest środowiskiem rozszerzalnym, tj. każdy użytkownik może zainstalować oficjalne lub stworzyć własne rozszerzenia, które dodają kolejne funkcjonalności w celu ułatwienia pracy nad projektem w Visual Studio.

Istnieje wiele wersji Visual Studio a każda z jego wersji jest wydana w trzech edycjach:

* Community – darmowa edycja, z której można skorzystać w projektach niekomercyjnych.
* Professional – edycja przeznaczona do projektów komercyjnych.
* Enterprise – edycja, która rozszerza edycję Professional o dodatkowe narzędzia.

Do utworzenia systemu wspomagania obserwacji astronomicznych skorzystano z Visual Studio 2019 Community. Narzędzia, które dostarcza ta wersja oraz jej przeznaczenie dla projektów niekomercyjnych powoduje, że jest to dobry wybór w celu utworzenia wspomnianego sytemu.

## 2.3 Języki programowania

Ze względu na wybór środowiska .NET Core ograniczony został wybór języków programowania do wspiernych przez to środowisko języków. Najbardziej wspieranymi językami przez .NET Core są języki programowania C# oraz F#, które zostały wykorzystane do stworzenia omawianego systemu.

### 2.3.1 C#

Pierwszym, w największym stopniu wykorzystanym językiem programowania do stworzenia projektu był język C#, który odpowiada za logikę oprogramowania oraz interfejs użytkownika.

C# jest językiem wieloparadygmatowym, który głównie wspiera paradygmat obiektowy, dzięki czemu projektowanie oprogramowania w ramach tego języka nie jest ograniczona do jednego paradygmatu co pozwala na zastosowanie wszelkich zalet pochodzących od istniejących paradygmatów, szczególnie dwóch najbardziej wykorzystanych w projekcie czyli paradygmat obiektowy oraz funkcyjny.

Kolejną ważną zaletą tego języka jest to, że pamięć operacyjna, którą ten język alokuje, jest kontrolowana przez środowisko uruchomieniowe (tzw. Garbage Collector) dzięki czemu podczas projektowania mniej uwagi trzeba poświęcić na alokowanie i zwalnianie pamięci co przyspiesza pracę z kodem.

C# jest również językiem silnie typowanym co oznacza, że na poziomie kompilacji kodu wiadoma jest potencjalna niezgodność wykorzystanych typów, pozwala to na bezpieczne projektowanie oprogramowania dające mniejsze szanse na wystąpienie defektu, jest to więc duża zaleta tego języka programowania.

Język C# jest kompilowany do języka Common Intermediate Language (CIL), który jest kodem pośrednim w środowisku .NET Core. Z tego powodu z poziomu kodu C# istnieje możliwość bezpośredniego wykorzystania kodu z innych języków, które też kompilowane są do CIL.

!!!

Ze względu na wyżej wymienione cechy, C# jest głównie wykorzystanym językiem w systemie wspomagania obserwacji astronomicznych. Jego różnorodność, łatwość zarządzania pamięcią, bezpieczeństwo oraz bycie częścią platformy .NET Core ułatwia pracę nad oprogramowaniem i gwarantuje, że oprogramowanie będzie niezawodne.

### 2.3.2 F#

Drugim wykorzystanym językiem został język F#, w którym zaimplementowane zostały algorytmy związane z dziedziną oprogramowania czyli z obliczeniami astronomicznymi.

F#, jak C#, jest również językiem wieloparadygmatowym jednakże głównym paradygmatem wspieranym przez ten język jest paradygmat funkcyjny. Paradygmat ten pozwolił na zaimplementowanie algorytmów jako czyste funkcje, które nie wymagają być zaimplementowanymi w danej klasie obiektu, tak jak jest to w języku C#. Paradygmat ten umożliwia również napisanie krótszego i zgrabniejszego kodu, w porównaniu do C#, który jest bardziej opisowy, matematyczny i zrozumiały dla człowieka.

Język ten należy do środowiska .NET Core i jest kompilowany do CIL, dzięki czemu zaimplementowane algorytmy obliczeniowe w nim mogą zostać wykorzystane bezpośrednio w kodzie języka C#. W ten sposób w ramach jednej warstwy, w tym przypadku odpowiadającej za obliczenia astronomiczne, można wykorzystać inny język programowania aby wykorzystać jego potencjał i nie doświadczyć żadnych problemów wynikających z użycia dwóch języków w tym samym oprogramowaniu takich jak konwersja typów dla danego środowiska innego języka programowania.

## 2.4 Windows Presentation Foundation

W środowisku .NET Core jest wiele różnych frameworków wspierających tworzenie interfejsu użytkownika dla różnych systemów oraz rodzajów aplikacji. Jednym z tych frameworków celujących w aplikacje desktopowe na systemy Windows jest Windows Presentation Foundation (WPF). Wspiera on wytwarzanie różnego rodzaju funkcjonalności desktopowych aplikacji w tym modelu aplikacji, zasobów, kontrolek, grafik, layoutu, wiązania danych i bezpieczeństwa.

### 2.4.1 Język opisu interfejsu użytkownika w WPF - XAML

Do opisu interfejsu użytkownika framework WPF wykorzystuje się język XAML (Extensible Application Markup Language). Oparty on jest na języku XML rozwinięty o dodatkowe elementy pozwalające na dokładny i czytelny opis interfejsu użytkownika dzięki czemu w łatwy i szybki sposób można zdefiniować rozmieszczenie oraz wygląd różnych elementów interfejsu graficznego.

XAML udostępnia również wiązanie danych, które pozwala na zdefiniowanie relacji między źródłem danych, a odbiorcą danych dzięki czemu w łatwy sposób można oddzielić logikę oprogramowania od widoku użytkownika. W relacji tej możliwe jest zdefiniowanie czy dane mają być odczytane tylko raz ze źródła lub czy źródło danych może być modyfikowane z poziomu widoku użytkownika czyli odbiorcy. W wiązaniu danych istnieje również możliwość dodania konwersji danych pochodzących ze źródła do odbiorcy i odwrotnie co pozwala na umieszczenie logiki, która jest tylko i wyłącznie związana z prezentacją danych więc nie trzeba w takiej sytuacji zmieniać logiki oprogramowania.

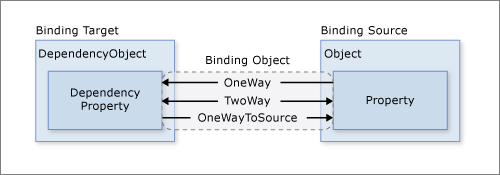


Diagram 1 Kierunek przepływu danych w wiązaniu danych.

Istotną zaletą XAML są istniejące możliwości utworzenia stylów, które określają wygląd oraz sposób prezentacji różnych elementów interfejsu graficznego. Takie style są do wielokrotnego wykorzystania i co za tym idzie można wykorzystywać ciągle te same style i szablony bez ponownego ich definiowania. Pozwala to na zachowanie spójności w wygodzie i możliwie sprawnej modyfikacji wyglądu danej grupy elementów interfejsu z tym samym stylem.

Styl określający rozmiar, kolor i pogrubienie tekstu

Przycisk z napisem OK

Etykieta z napisem Potwierdź

Diagram 2 Przykładowe wykorzystanie istniejącego stylu w dwóch różnych kontrolkach.

We współczesnych interfejsach użytkownika często wykorzystywane są animacje do polepszenia odbioru wyglądu danego interfejsu. Z tego powodu w XAML i w WPF udostępniono możliwość opisania animacji w formie nazywanej Storyboard. Taka forma udostępnia wiele właściwości związanych z przebiegiem animacji dzięki czemu dana animacja jest łatwa do kontrolowania oraz tworzenie jej jest szybkie i proste.

### 2.4.2 Silnik graficzny WPF

Ze względu na coraz większe zaawansowanie interfejsów użytkownika WPF wspiera nie tylko animacje i grafiki 2D ale również 3D. Takie animacje i grafiki jednak mogą mocno obciążyć system operacyjny. Ze względu na to WPF automatycznie wykorzystuje przyspieszenie sprzętowe (ang. Hardware Acceleration) korzystając z Microsoft DirectX API, które udostępnia interfejs programistyczny do kontrolowania grafiki w systemie Windows. Pozwala to na wykorzystanie karty graficznej co polepsza wydajność aplikacji WPF jak i wyświetlanie animacji oraz grafik.

### 2.4.3 Wieloplatformowość

Windows Presentation Foundation powstał w celu projektowania aplikacji desktopowych dla systemu Windows i nie przewidywano aby framework ten mógł być wykorzystany również na innych systemach operacyjnych. Fakt ten powoduje, że wykorzystanie zdefiniowanego interfejsu użytkownika w WPF nie jest możliwe w innych systemach operacyjnych niż Windows. Współcześnie jest to duża wada WPF ze względu na coraz szersze wykorzystanie innych systemów operacyjnych takich jak Linux i Mac OS niż system Windows.

## 2.5 SQLite i DB Browser

System wspomagania obserwacji astronomicznych wyświetla wiele obiektów na mapie więc musi to robić szybko. Najlepszym rozwiązaniem w takiej sytuacji jest aby użytkownik miał lokalną bazę danych, która przechowuje wszystkie potrzebne informacje do generowania mapy.

SQLite jest systemem bazodanowym do zarządzania relacyjną bazą danych, która może być przechowywana w jednym pliku, jest to więc narzędzie, które najbardziej nada się do wykorzystania w systemie wspomagania obserwacji astronomicznych.

Do utworzenia i zarządzania taką bazą została wykorzystana desktopowa aplikacja DB Browser. Pozwala ona na tworzenie baz danych i ich tabel oraz pozwala na pisanie kwerend.

# 3. Podstawy teoretyczne zagadnień astronomicznych

W pracy niniejszego tematu wykorzystano podstawowe zagadnienia astronomiczne, związane z położeniem ciał niebieskich na sferze niebieskiej względem danego miejsca i czasu oraz z ich właściwościami.

## 3.1 Czas w astronomii

Ważną częścią określenia pozycji danego obiektu na sferze niebieskiej jest czas obserwacji. W astronomii używanych jest wiele definicji czasu dla różnych zastosowań. Dlatego w niniejszej pracy skupiono się na najważniejszych dla tego tematu definicjach.

### 3.1.1 Dni juliańskie

Jest to data używana w astronomii i jest przedstawiana w postaci liczby dni w postaci dziesiętnej, które upłynęły według kalendarza juliańskiego od fundamentalnej epoki czyli 1 stycznia 4713 p.n.e, godziny 12 czasu uniwersalnego (UTC). Przedstawianie daty w takiej formie ułatwia obliczenia astronomiczne ponieważ w tym przypadku korzysta się tylko z jednej konkretnej liczby. Oznacza się ją symbolem **JD.**

### 3.1.2 Lokalny czas gwiazdowy

Czas ten wyznacza się tempem rotacji sfery niebieskiej. Z definicji jest kątem godzinnym punktu równonocy wiosennej i jest zawsze równy rektascensji obiektu astronomicznego, który znajduje się w danej chwili w południku lokalnym (więcej o tym w podrozdziale [współrzędne równikowe](#_2.2.2_Współrzędne_równikowe)). Symbolem tego czasu jest **LST** (ang. Local Sidereal Time). Lokalny czas gwiazdowy ma charakter lokalny, więc w różnych miejscach na Ziemi nie jest taki sam.

## 3.2 Układy współrzędnych astronomicznych

Istnieje kilka układów współrzędnych używanych w astronomii, które mają pomiędzy sobą ścisłą matematyczną relację. Każdy układ współrzędnych przydaje się do różnych zastosowań takich jak określenie lokalnego położenia ciała niebieskiego względem obserwatora lub ustalenie położenia jego w galaktyce.

### 3.2.1 Współrzędne horyzontalne

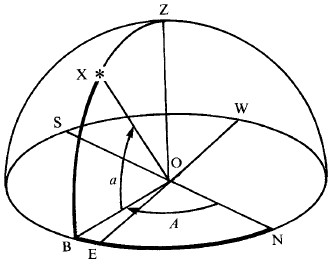
Jest to najprostszy układ współrzędnych w astronomii i najbardziej naturalny z punktu widzenia obserwatora.

Płaszczyzna w tym układzie współrzędnych jest styczna do powierzchni Ziemi oraz przechodzi przez obserwatora. Przecięcie tej płaszczyzny ze sferą niebieską jest horyzontem. Punkt położony na sferze dokładnie nad obserwatorem nazywany jest zenitem.

Jedną ze współrzędnych tego układu jest wysokość, a, która mierzona jest od zenitu do nadiru (punkt położony dokładnie pod obserwatorem). Wysokość zawiera się w zakresie [-90°; +90°], gdzie wartości dodatnie znajdują się nad horyzontem a ujemne poniżej.

Drugą współrzędną jest azymut, A, która oznacza kąt przechodzący na płaszczyźnie od ustalonego punktu na horyzoncie do danego obiektu. Azymut wyrażany jest głównie liczbami od 0° do 360°. Punktem początkowym zazwyczaj jest kierunek północny.

Wadą tego układu współrzędnych jest fakt, że współrzędne te zmieniają się wraz z upływem czasu. Stosuje się więc ten układ współrzędnych w celu określenia położenia danego obiektu na sferze niebieskiej w danym konkretnym czasie i względem danego punktu obserwacyjnego.



Rysunek 1 Współrzędne horyzontalne.

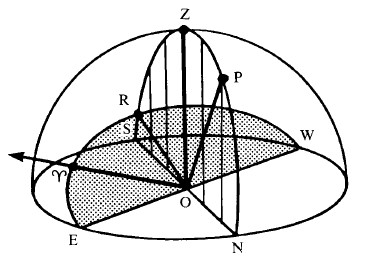
### 3.2.2 Współrzędne równikowe

Aby określić stałe położenie obiektu na sferze niebieskiej korzysta się ze współrzędnych równikowych. Kierunek osi obrotu Ziemi jest niemal stały. Z tego powodu niezmienne jest również położenie prostopadłej do niej płaszczyzny równika. Jest to więc wygodna płaszczyzna podstawowa dla układu współrzędnych niezależnego od czasu i pozycji obserwatora.

Warto jednak zaznaczyć, że pozycja obiektów w przestrzeni kosmicznej oraz oś obrotu Ziemi zmieniają się wraz z upływem dłuższego czasu dlatego współrzędne równikowe dla obiektów spoza Układu Słonecznego są aktualizowane co 50 lat. Aktualnym katalogiem jest katalog o nazwie J2000.0, który został utworzony w 2000 roku.

Pierwszą współrzędną tego układu jest deklinacja, δ, która oznacza odległość kątową od płaszczyzny równika. Podobnie jak wysokość we współrzędnych horyzontalnych, deklinacja zawiera się w zakresie [-90°; +90°], gdzie wartości dodatnie znajdują się nad przecięciem płaszczyzny równika a ujemne poniżej.

W celu określenia drugiej współrzędnej potrzebny jest punkt, względem którego będzie ta współrzędna określana. Za ten punkt przyjmuje się *punkt równonocy wiosennej*, który niegdyś znajdował się w gwiazdozbiorze Barana, dlatego oznacza się go symbolem gwiazdozbioru - ϒ. Dzięki temu można zdefiniować potrzebną współrzędną – rektascensję, α. Jest mierzona odwrotnie do wskazówek zegara i przyjęło się, że jest mierzona od 0h do 24h.



Rysunek 2 Współrzędne równikowe.

### 3.2.3 Inne współrzędne

Oprócz podanych wyżej układów współrzędnych, w astronomii są zdefiniowane jeszcze dwa układy współrzędnych: ekliptyczny oraz galaktyczny.

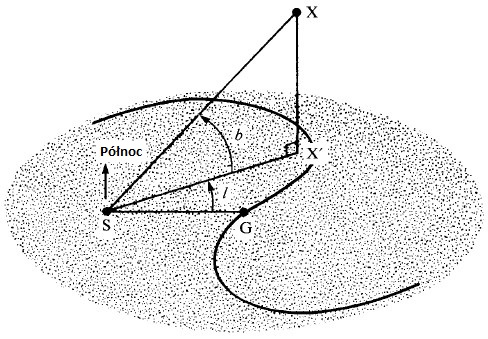
W pierwszym z nich płaszczyzną podstawową jest koło wielkie, które zataczane jest na sferze niebieskiej przez Słońce w jego ruchu rocznym – odbicie orbity Ziemi. Układ ten określają dwie współrzędne czyli szerokość ekliptyczna, β, oraz długość ekliptyczna, λ. Wykorzystywany jest on do określenia pozycji obiektów w Układzie Słonecznym.

Diagram, engineering drawing

Description automatically generated

Rysunek 3 Współrzędne ekliptyczne.

Współrzędne galaktyczne mają płaszczyznę podstawową zawartą w płaszczyźnie Drogi Mlecznej. Słońce leży bardzo blisko tej płaszczyzny dlatego umieszcza się je na początku tego układu. Układ ten składa się z długości galaktycznej, l, oraz z szerokości galaktycznej, b. Współrzędne te określają pozycję obiektów w galaktyce Droga Mleczna.



Rysunek 4 Współrzędne galaktyczne.

## 3.3 Konwersje między współrzędnymi

Często w astronomii przydaje się konwersja z jednego układu współrzędnych na drugi. Przykładowo znając współrzędne równikowe danego obiektu możemy przekonwertować je na współrzędne horyzontalne dzięki czemu możliwe będzie określenie pozycji obiektu na lokalnym niebie.

Konwersje między współrzędnymi równikowymi a pozostałymi trzema są względnie jasne. Z tego właśnie powodu konwersje wykonywane są najczęściej poprzez układ współrzędnych równikowych. Zilustrowane to jest na Diagramie 3.

Ekliptyczne

Galaktyczne

Horyzontalne

Równikowe

Diagram 3 Konwersje współrzędnych.

Istnieje też alternatywny sposób konwersji pomiędzy wspomnianymi współrzędnymi. Zamiast korzystać z bezpośrednich formuł dla każdej konwersji można wykorzystać metodę macierzową ogólnego przekształcenia współrzędnych. Ten sposób daje możliwość bezpośredniej konwersji między jednym systemem a drugim jednak jest bardziej skomplikowany.

### 3.3.1 Konwersja współrzędnych równikowych do horyzontalnych

Relacje pomiędzy kątem godzinnym, H, deklinacją, δ, azymutem, A, oraz wysokością, a, są następujące:

gdzie φ jest szerokością geograficzną obserwatora. Do konwersji potrzebny jest kąt godzinny, H, który oblicza się wzorem:

gdzie LST jest lokalnym czasem gwiazdowym, a α jest rektascensją.

### 3.3.2 Konwersja współrzędnych horyzontalnych do równikowych

W tym przypadku problem jest odwrotny do tego w przypadku konwersji współrzędnych równikowych na horyzontalne. Relacje między wysokością, a, azymutem, A, deklinacją, δ oraz kątem godzinnym H są następujące:

gdzie φ ponownie jest szerokością geograficzną obserwatora. Tak jak w konwersji ze współrzędnych równikowych na horyzontalne pojawia się kąt godzinny, który można wykorzystać do obliczenia rektascensji poprzez przekształcenie wzoru na kąt godzinny:

### 3.3.3 Konwersje pozostałych współrzędnych

W pracy niniejszego tematu nie wykorzystano współrzędnych ekliptycznych oraz galaktycznych dlatego omówienie konwersji między nimi zostaje pominięte.

## 3.4 Właściwości obiektów astronomicznych

Na sferze niebieskiej znajduje się wiele obiektów astronomicznych i każdy z nich należy do pewnego rodzaju tych obiektów oraz posiada wiele różnych cech. Z tego powodu obiekty astronomiczne są pogrupowane względem swojego rodzaju oraz zdefiniowano wiele właściwości opisujących je.

W temacie pracy skupiono się na gwiazdach zarejestrowanych w najpopularniejszych powszechnych katalogach gwiazd dlatego opisane zostały tylko właściwości związane z tymi obiektami.

### 3.4.1 Odległość od Słońca

Każdy obiekt na sferze niebieskiej jest w pewnej odległości od Ziemi. W przypadku obiektów międzygwiazdowych odległość ta mierzona jest względem Słońca ze względu na jego względnie stałą pozycję w Układzie Słonecznym. Główne jednostki do określania odległości w przestrzeni kosmicznej są następujące:

* Jednostka astronomiczna (AU, ang. astronomical unit) – wynosi dokładnie 149 597 870 700 metrów i jest przybliżonym średnim dystansem Ziemi od Słońca.
* Rok świetlny (ly, ang. light year) – wynosi około 9,5 bilionów kilometrów.
* Parsek (pc, ang. parsec) – wynosi około 3.26 lat świetlnych.

### 3.4.2 Wielkość gwiazdowa

Do określenia blasku gwiazd (oraz innych obiektów na niebie) służy wielkość gwiazdowa, która jest jednostką miary wyrażaną w magnitudo. Wartości jej mogą być dodatnie oraz ujemne gdzie im mniejsza wartość tym większy blask gwiazdy. Wielkość gwiazdową generalnie dzieli się na dwa rodzaje:

* Wielkość obserwowana – wielkość gwiazdowa danego obiektu, która zależy od jego jasności i odległości od Ziemi.
* Wielkość absolutna – wielkość gwiazdowa obiektu obserwowanego z ustalonej odległości przy założeniu braku pochłaniania światła w przestrzeni międzygwiazdowej. Dla obiektów międzygwiezdnych odległość ta została przyjęta jako 10 parseków.

### 3.4.3 Typ widmowy

Jest to klasyfikacja gwiazd oparta na ich widmie światła. Dzięki tej klasyfikacji można określić kolor danej gwiazdy. Większość gwiazd na niebie można podzielić na siedem podstawowych typów:

|  |  |
| --- | --- |
| Typ | Kolor |
| O | Błękitny |
| B | Niebiesko-biały |
| A | Biały |
| F | Żółto-biały |
| G | Żółty |
| K | Pomarańczowy |
| M | Czerwony |

Tabela 1 Typy widmowe i ich kolory.

# 4. Projekt systemu

Głównym i najważniejszym celem systemu jest wyświetlanie interaktywnej mapy nieba. Widocznych obiektów na sferze niebieskiej jest bardzo wiele jak również możliwości rozwoju tego systemu. Z tego powodu projekt systemu został stworzony uwzględniając te aspekty skupiając się na optymalizacji wyświetlania mapy oraz architekturze pozwalającej na łatwe dołączenie kolejnych funkcjonalności do systemu.

## 4.1 Analiza funkcjonalności

Funkcjonalności są stwierdzeniami dotyczącymi usług dostarczanych przez system, sposobu realizacji systemu na określone dane wejściowe i tego, jak powinien się zachowywać w pewnych sytuacjach. W niektórych przypadkach wymagania funkcjonalne mogą też jawnie stwierdzać, czego system nie może robić.

W systemie wspomagania obserwacji astronomicznych zdefiniowanych takich wymagań jest kilka i główne z nich zostały omówione w tym rozdziale.

### 4.1.1 Przechowywanie i pobieranie obiektów z bazy danych

System wspomagania obserwacji astronomicznych wyświetla mapę nieba z wieloma obiektami. Do jej prawidłowego i zoptymalizowanego działania wykorzystuje odpowiednią bazę danych, która przechowuje wiele obiektów z informacjami na ich temat i pozwoli na szybki odczyt. System potrafi również określić jakie konkretnie obiekty go interesują z bazy danych.

### 4.1.2 Wyświetlanie obiektów na mapie nieba

System wyświetla kołową, dwuwymiarową mapę zawierającą obiekty znajdujące się na niebie. Każdy zawarty w niej obiekt ma określony kształt, kolor oraz odpowiednią pozycję. Na mapie pokazane są kierunki geograficzne północ, wschód, południe oraz zachód. W środku mapy ukazana jest siatka ułatwiająca odniesienie pozycji danego obiektu na rzeczywistej sferze niebieskiej.

### 4.1.3 Interaktywność i sterowanie mapy

Mapa systemu jest interaktywna czyli użytkownik ma możliwość obrotu mapy, przesuwania się po niej oraz przybliżenia i oddalenia od danego obszaru na niej.

### 4.1.4 Wyświetlanie informacji o danym obiekcie

System po wybraniu danego obiektu wyświetlonego na mapie pokazuje informacje na temat tego właśnie obiektu w panelu bocznym. Użytkownik może odznaczyć wybrany obiekt aby schować wyświetlone na jego temat informacje.

### 4.1.5 Ustawienie lokalizacji

Użytkownik może wprowadzić do systemu współrzędne geograficzne określające pozycję, z której obserwowane jest niebo. Współrzędne wyrażone są w stopniach gdzie szerokość geograficzna jest dodatnia dla szerokości na północ od równika a ujemna na południe od niego oraz gdzie długość geograficzna jest dodatnia dla długości na wschód od południka zerowego a ujemna na zachód od niego. Użytkownik nie może podać współrzędnych wykraczających poza ich zakres, czyli szerokość geograficzna nie może być większa niż 90 stopni i nie mniejsza niż -90 stopni oraz długość geograficzna nie może być większa niż 180 stopni i nie mniejsza niż -180 stopni.

### 4.1.6 Ustawienie czasu

System pozwala użytkownikowi na ustawienie daty oraz czasu obserwacji, czyli użytkownik może osobno określić rok, miesiąc, dzień oraz godzinę, minutę i sekundę. Użytkownik ma możliwość również automatycznego wczytania lokalnej daty i czasu komputera.

### 4.1.7 Upływ i sterowanie czasem

W systemie możliwe jest dla użytkownika sterowanie upływem czasu poprzez uruchomienie lub zatrzymanie jego upływu, określenie kierunku jego upływu (w przyszłość lub przeszłość), oraz prędkości z jaką czas ma upływać. Użytkownik określa prędkość upływu czasu wskazując jaka część daty lub czasu ma się zmieniać co sekundę, np. o miesiąc lub o godzinę w wyznaczonym kierunku.

### 4.1.8 Aktualizacja mapy

System przy każdej zmianie lokalizacji oraz czasu aktualizuje mapę nieba aby wyświetlała odpowiednie obiekty odpowiednich dla wybranych ustawień.

### 4.1.9 Ocena warunków pogodowych

W systemie użytkownik może wprowadzić według swoich własnych obserwacji dane cechy pogodowe, na podstawie których system oceni warunki do prowadzenia obserwacji astronomicznych w godzinach nocnych. Na potrzeby tej funkcjonalności zdefiniowane zostały uwzględniane cechy pogodowe i ich skala ocen. Wszystkie wartości wynikające z danych wprowadzonych przez użytkownika są do siebie dodawane w ten sposób uzyskując ocenę warunków pogodowych gdzie wartości mniejsze od 0 oznaczają złe warunki a wartości większe od 0 oznaczają dobre warunki do obserwacji.

|  |  |  |
| --- | --- | --- |
| Nazwa | Wartości | Opis |
| Wysokość obserwacji | -2 poniżej 0 m n.p.m -1 poniżej 50 m n.p.m 0 poniżej 100 m n.p.m +1 poniżej 200 m n.p.m +2 wyżej lub równo 200 m n.p.m | To kryterium uwzględnia wysokość położenia obserwacji w metrach nad poziomem morza |
| Zachmurzenie | -2 dla oceny 1 -1 dla oceny 2 0 dla oceny 3 +1 dla oceny 4 +2 dla oceny 5 | Użytkownik ocenia zachmurzenie w skali od 1 do 5 gdzie im wartość jest większa tym mniejsze jest zachmurzenie |
| Sztuczne oświetlenie | -2 dla oceny 1 -1 dla oceny 2 0 dla oceny 3 +1 dla oceny 4 +2 dla oceny 5 | Użytkownik ocenia jak mocne jest sztuczne oświetlenie w miejscu obserwacji w skali od 1 do 5 gdzie im wartość jest większa tym mniejsze jest sztuczne oświetlenie |
| Widoczność Księżyca | Wartości prawda lub fałsz -2 dla wartości prawda +2 dla wartości fałsz | To kryterium uwzględnia czy Księżyc jest w czasie obserwacji widoczny na niebie. |

Tabela 2 Kryteria oceny warunków pogodowych w miejscu obserwacji.

## 4.2 Architektura

Aby zbudować niezawodny i łatwo rozszerzalny system należy taki system zaplanować. Jednym z pierwszych kroków do zaplanowania takiego systemu jest zaprojektowanie architektury. Projekt architektoniczny opisuje jak zorganizować system oprogramowania i skupia się na projektowaniu ogólnej struktury tego systemu. Architektura musi być dobrze zaprojektowana ponieważ przyszłe zmiany w architekturze są bardzo pracochłonne gdzie wiele komponentów może zależeć od siebie.

Do tej pory stworzonych zostało wiele różnych systemów informatycznych i tworzenie ich wiązało się z wymyśleniem nowych rozwiązań oraz sposobów projektowania oprogramowania. Z tego powodu aby ponownie wykorzystać wiedzę z tworzenia istniejących już systemów udostępniono tzw. wzorce architektoniczne, które opisują jak architektura danego systemu powinna zostać zaprojektowana. Każdy wzorzec ma swoje zalety i wady więc wybór zależy od tego jaki cel jest tworzonego oprogramowania.

### 4.2.1 Przegląd wzorców architektonicznych

W tym podrozdziale przedstawione są powszechnie używane wzorce architektoniczne.

|  |  |
| --- | --- |
| Nazwa | Opis |
| Repozytorium | Komponenty architektury nie wchodzą ze sobą w bezpośrednią interakcję, a jedynie za pomocą repozytorium, w którym przetrzymywane są wszystkie dane. |
| Warstwowa | Dzieli cały system na warstwy, które są związane z funkcjonalnościami przypisanymi do nich. Warstwa niżej udostępnia swoje usługi warstwie wyższej, nie odwrotnie. |
| Klient-serwer | System jest zbiorem usług, a każda z nich jest dostarczana przez osobny serwer. Klient to użytkownik, który wymaga dostępu do serwerów aby skorzystać z danych usług. |
| MVC (Model-View-Controller) | Oprogramowanie dzielone jest na trzy komponenty, które ze sobą współpracują. Komponent Model zarządza danymi i operacjami z nimi powiązanymi. View definiuje jak prezentowane są dane dla użytkownika. Controller zarządza interakcja z użytkownikiem i przekazuje te interakcje do pozostałych komponentów. |

Tabela 3 Wzorce architektoniczne.

### 4.2.2 Wybór architektury

Do utworzenia systemu wspomagania obserwacji astronomicznych użyto frameworka WPF, który wspiera architekturę MVVM (Model-View-ViewModel), która jest rozwinięciem architektury MVC. Dlatego architektura zastosowana w przypadku opisywanego systemu jest architekturą MVVM. Architektura ta pozwala na oddzielenie widoku (interfejsu użytkownika) od logiki oprogramowania, a wiązanie danych wspierane przez WPF ułatwia komunikację interfejsu z logiką niezależnie od implementacji. Dzięki temu oprogramowanie jest łatwiejsze w przypadku rozbudowywania go o nowe funkcjonalności i prezentację widoku.

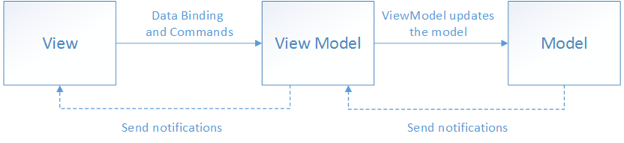


Diagram 4 Wzorzec MVVM.

### 4.2.3 Projekt architektury systemu

W tym podrozdziale za pomocą diagramów UML przedstawiony zostanie projekt architektury omawianego systemu. Diagramy UML służą do modelowania różnego rodzaju systemów, w których opisane są z czego składają się poszczególne elementy systemu i jakie są między tymi elementami relacje.

*CelestialObject*

Graphical user interface

Description automatically generated with medium confidence

Diagram 5 Klasa CelestialObject, jej ViewModel oraz StarObject.

W celu reprezentacji obiektu astronomicznego została zaprojektowana jedna klasa abstrakcyjna o nazwie *CelestialObject* zawierająca wszystkie podstawowe dane obiektu astronomicznego, czyli: jego nazwa, współrzędne równikowe, współrzędne horyzontalne, wielkość gwiazdowa obserwowana oraz dystans do Ziemi. Definiuje również właściwość Id, które pozwala na bezpośrednią identyfikację obiektu.

W przypadku dodania nowego obiektu astronomicznego do wyświetlania na mapie, np. planety, wystarczyłoby dodanie kolejnego obiektu z informacjami na jego temat, który dziedziczyłby po klasie *CelestialObject.*

Klasa *CelestialObjectViewModel* reprezentuje widok klasy *CelestialObject*. Odzielone są tutaj dane modelu od danych potrzebnych warstwie widoku. Widok modelu tej klasy definiuje podstawowe właściwości określające pozycję obiektu, jego kolor, wielkość oraz zawiera informacje czy dany obiekt jest zaznaczony na mapie i czy jest widoczny. Kolekcja *CelestialObjectInfoCollection* zawiera klasy typu *ICelestialObjectInfoViewModel*, które generowane są względem typu modelu, który reprezentują i zawierają informacje na temat konkretnych właściwości reprezentowanego obiektu astronomicznego.

Timeline

Description automatically generated

Diagram 6 Interfejs ICelestialObjectInfo i jej wersja generyczna.

Do utworzenia instancji typu *ICelestialObjectInfoViewModel* zastosowano wzorzec projektowy o nazwie „Budowniczny”. Klasa *CelestialObjectInfoBuilder* jest klasą abstrakcyjną i zawiera jedną definicję abstrakcyjnej metody, która zwraca kolekcję obiektów *ICelestialObjectInfo*. Dzięki temu tworzenie kolekcji z informacjami o danym obiekcie jest elastyczne i łatwe w potencjalnej modyfikacji.

A picture containing chart

Description automatically generated

Diagram 7 Klasa CelestialObjectInfoViewModelCollectionBuilder.

*ICelestialObjectCollection*

Text

Description automatically generated

Diagram 8 Interfejs ICelestialObjectCollection.

Jest to interfejs definiujący kolekcję do przechowywania obiektów klasy *CelestialObject.* Interfejs ten dziedziczy po wbudowanym w .NET Core interfejsie *ICollection<T>*, aby był zgodny ze standardem środowiska. Interfejs definiuje trzy zdarzenia, które wywoływane są w przypadku akcji powiązanej z nimi. Zdarzenie Added wywoływane jest kiedy do kolekcji zostaną dodane obiekty, Removed kiedy zostaną usunięte, a Cleared kiedy wszystkie obiekty zostaną usunięte.

Dzięki zastosowaniu interfejsu nie ma ograniczenia co do zastosowania rodzaju kolekcji co pozwala na łatwą zmianę implementacji.

*SwoaManager*

Diagram

Description automatically generated

Diagram 9 Klasa SwoaManager i jej widok modelu.

Klasa *SwoaManager* jest główną klasą zawierającą wszystkie usługi związane z systemem wspomagania obserwacji astronomicznych i przyjmuje rolę mediatora, który przekazuje między nimi potrzebne informacje. W tej implementacji *SwoaManager* zawiera dwie usługi, które będą omówione w dalszej części, czyli: *CelestialObjectManager* oraz *SwoaWeatherManagerViewModel.*

*CelestialObjectManager*

Diagram

Description automatically generated with medium confidence

Diagram 10 Klasa CelestialObjectManager.

Klasa *CelestialObjectManager* i jej widok modelu służą do zarządzania i przechowywania załadowanych do systemu obiektów astronomicznych. Model klasy przechowuje te obiekty a widok modelu aktualizuje ich pozycje na mapie w odpowiedzi na zmiany wielkości mapy. Aktualizacja pozycji obiektów astronomicznych jest wykonywana asynchronicznie co oznacza, że wykorzystywany jest inny wątek dzięki czemu wątek główny nie jest obciążany i interfejs użytkownika nie blokuje się.

TimeMachine

Diagram

Description automatically generated

Diagram 11 Klasa TimeMachine i jej widok modelu.

Do kontrolowania upływu czasu w systemie oraz określania pozycji obserwacji służy klasa *TimeMachine* i jej widok modelu. Klasa *TimeMachine* zawiera dane wprowadzone przez użytkownika o lokalizacji punktu obserwacyjnego, czasie oraz tego w jakim kierunku i jak szybko upływać ma czas. Do klasy tej należy podać serwis *IDateTimeService*, który służy do pobierania aktualnej daty, co pozwala użytkownikowi na automatyczne wprowadzenie jej do systemu. *TimeMachine* nie kontroluje bezpośrednio kolekcji obiektów astronomicznych, a tylko informuje kiedy aktualizacja i dla jakich danych powinna nastąpić.

SwoaDb

Text

Description automatically generated with low confidence

Diagram 12 Klasa SwoaDb.

W celu pobierania danych z bazy danych obiektów astronomicznych służy klasa abstrakcyjna *SwoaDb*. Klasa ta jest abstrakcyjna co pozwala na utworzenie implementacji odpowiadających konkretnej bazie danych więc uniezależnia to system od zastosowanej bazy danych. Zawiera ona definicje abstrakcyjnych metod pozwalających na wysłanie zapytania do bazy danych oraz określenie, których obiektów o podanym id ma nie pobierać w celach optymalizacyjnych. Do *SwoaDb* można przekazać tzw. token anulujący, który zawiera informacje czy pobieranie obiektów z bazy ma być przerwane.

*SwoaDb* pobierając obiekty astronomiczne tworzy i zwraca instancje klasy *SwoaDbRecord*. Są to proste obiekty zawierające konkretne nieprzetworzone właściwości pobrane z bazy danych.

A picture containing table

Description automatically generated

Diagram 13 Klasa SwoaDbRecord.

CelestialObjectsLoader

Graphical user interface, text, application

Description automatically generated

Diagram 14 Klasa CelestialObjectsLoader.

*CelestialObjectsLoader* jest specjalną klasą służącą do wypełniania kolekcji obiektów astronomicznych. Do pobierania obiektów astronomicznych wykorzystuje klasę *SwoaDb*, a dane do obliczania ich pozycji na lokalnym niebie (mowa o współrzędnych horyzontalnych) pobiera z klasy *TimeMachine*. Pobieranie i przetwarzanie danych obiektów astronomicznych jest wykonywane asynchroniczne a token anulujący pozwala na przerwanie tego procesu, np. kiedy użytkownik zmieni dane w klasie *TimeMachine* przed jego zakończeniem.

SwoaWeatherManager

Chart

Description automatically generated

Diagram 15 Klasa SwoaWeatherManager i jej widok modelu.

Funkcjonalnością związaną z oceną warunków pogodowych zajmuje się klasa *SwoaWeatherManager* oraz jej widok modelu. Na podstawie zdefiniowanych w niej właściwości obliczana jest ocena warunków pogodowych, która jest przechowywana we właściwości *WeatherCondition*. Widok modelu przy okazji definiuje właściwość *IsPanelShown*, która wskazuje czy panel w interfejsie jest widoczny.

## 4.3 Baza danych

Obiektów astronomicznych widzialnych z powierzchni Ziemi jest bardzo wiele a informacji o nich jest jeszcze więcej. Z tego powodu w systemie wspomagania obserwacji astronomicznych potrzebna jest baza danych, która przechowa potrzebne informacje na temat obiektów astronomicznych w taki sposób aby dane zajmowały jak najmniej miejsca a odczyt ich był szybki.

### 4.3.1 Źródło danych

W omawianej wersji systemu wspomagania obserwacji astronomicznych wyświetlane są jedynie gwiazdy na mapie. Wystarczającym źródłem zawierającym dane na temat gwiazd są katalogi hipparcosa, jasnych gwiazd oraz gwiazd położonych najbliżej Ziemi. Odliczając te gwiazdy, które powtarzają się w wymienionych katalogach jest ich w sumie prawie 120 000.

### 4.3.2 Projekt bazy danych

W bazie danych ze względu na to, że jedynie gwiazdy są wyświetlane w systemie obserwacji astronomicznych istnieje tylko jedna tabela *stars* zawierająca wszystkie potrzebne dane. Tabela *stars* definiuje 14 atrybutów:

Table

Description automatically generated

Diagram 16 Tabela ‘stars’ w bazie danych.

|  |  |
| --- | --- |
| nAZWA KOLUMNY | OPIS |
| ID | Klucz główny bazy danych. |
| Hip | Id gwiazdy w katalogu Hipparcosa. |
| Hd | Id gwiazdy w katalogu Henry’ego Drapera. |
| hr | Id gwiazdy w katalogu jasnych gwiazd |
| gl | Id gwiazdy w katalogu najbliższych gwiazd Ziemi. |
| bf | Id gwiazdy w katalogu Bayer-Flamsteed. |
| proper | Podstawowa nazwa gwiazdy, jeśli taką dana gwiazda ma. |
| ra | Rektascensja gwiazdy dla epoki J2000 |
| dec | Deklinacja gwiazdy dla epoki J2000 |
| sundist | Odległość gwiazdy od Słońca w latach świetlnych. |
| mag | Obserwacyjna wielkość gwiazdowa wyrażona w magnitudo. |
| absmag | Absolutna wielkość gwiazdowa wyrażona w magnitudo |
| spect | Typ widmowy gwiazdy. |
| Con | Skrótowa nazwa konstelacji, do której należy gwiazda. |

Tabela 4 Atrybuty tabeli stars.

W celach optymalizacyjnych zdefiniowano również dwie funkcje do wykorzystania w kwerendach wyciągających dane z bazy danych: *skycontains* oraz *blacklisted*. Funkcja *blacklisted* jest wykorzystywana głównie przez klasę *SwoaDb* opisaną w rozdziale[*Projekt Architektury Systemu*](#_3.2.3_Projekt_architektury)*.*

|  |  |  |
| --- | --- | --- |
| nAZWA FUNKCJI | ARGUMENTY | oPIS |
| SKYCONTAINS | Współrzędne równikowe,  data,  współrzędne geograficzne | Jeśli dla podanych argumentów obiekt astronomiczny jest nad horyzontem zwracana jest wartość *true*, w innym przypadku wartość *false.* |
| blacklisted | Id | Jeśli obiekt o podanym id zawarty jest w czarnej liście obiektów zwracana jest wartość *true*, w innym przypadku wartość *false*. |

Tabela 5 Funkcje bazy danych.

## 4.4 Implementacja

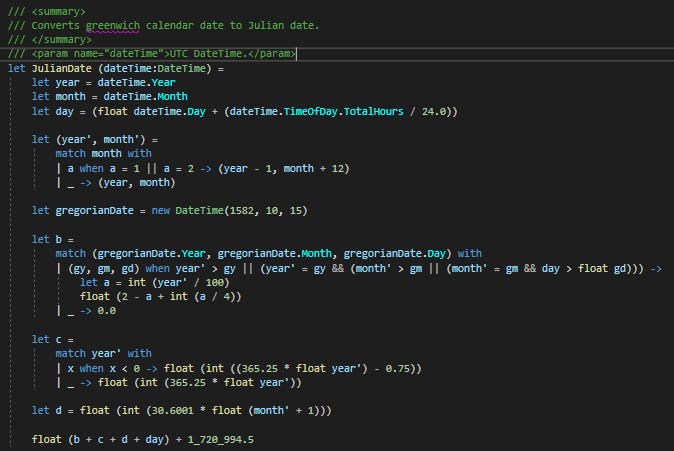
W tym rozdziale opisane będą najważniejsze części implementacji projektu systemu, w których wczytywane są obiekty astronomiczne z bazy danych oraz ich pozycje przy wykorzystaniu algorytmów astronomicznych.

### 4.4.1 Algorytmy astronomiczne

W rozdziale [*Podstawy teoretyczne zagadnień astronomicznych*](#_2._Podstawy_teoretyczne)przedstawione zostały wzory obliczeń potrzebnych do określenia obiektów astronomicznych na mapie sfery niebieskiej. Zastosowanie tych wzorów jednak bezpośrednio nie jest możliwe ze względu na to jak obliczane są daty oraz jak zaimplementowane są algorytmy trygonometryczne w systemach komputerowych. Wszystkie algorytmy astronomiczne w tym rozdziale zostały zaimplementowane zgodnie z książka !!!.

*Algorytm obliczający datę juliańską*

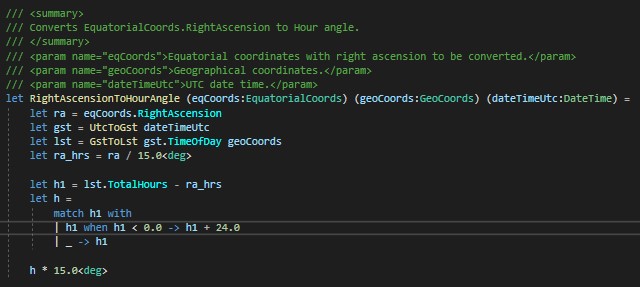
|  |
| --- |
| 1. Niech *y* będzie rokiem, *m* miesiącem a *d* dniem miesiąca czasu uniwersalnego (UT) 2. Jeśli *m* = 1 lub *m* = 2 to *y’* = *y* – 1 oraz *m’* = *m* + 12 w innym przypadku *y’* = *y* oraz *m’* = *m* 3. Jeśli data jest datą później niż 15.10.1582 (w gregoriańskim kalendarzu) *A* = int(*y’* / 100) *B* = 2 – *A* + int(*A* / 4) w innym przypadku *B* = 0 4. Jeśli *y’* < 0 to *C* = int((365,25 \* *y’*) – 0,75),  w innym przypadku *C* = int(365,25 \* *y’*) 5. *D* = int(30,6001 \* (*m’* + 1)) 6. *JD* = *B* + *C* + *D* + *d* + 1720994,5 |



Rysunek 3 Implementacja algorytmu obliczania daty juliańskiej w języku F#.

*Algorytm konwersji rektascensji do kąta godzinnego*

|  |
| --- |
| 1. Niech α będzie rektascensją w systemie dziesiętnym, *UT* będzie uniwersalną godziną dziesiętną dnia, *GDay* dniem, *GMonth* miesiącem, a *GYear* rokiem uniwersalnej daty 2. Znaleźć czas GST odpowiadający czasowi *UT* 3. Znaleźć czas LST odpowiadający czasowi GST 4. Od LST odjąć α. Dodać 24 jeśli wynik jest negatywny |



Rysunek 4 Implementacja algorytmu konwersji rekstascensji do kąta godzinnego w języku F#.

*Algorytm konwersji współrzędnych równikowych do współrzędnych horyzontalnych*

|  |
| --- |
| 1. Niech H będzie kątem godzinnym, δ deklinacją a φ szerokością geograficzną punktu obserwacyjnego 2. H’ = 15 x H 3. Obliczyć: 4. Jeśli jest ujemne to , w innym przypadku |

A screenshot of a computer

Description automatically generated with medium confidence

Rysunek 5 Implementacja algorytmu konwersji współrzędnych w języku F#.

### 4.4.2 Obliczanie pozycji obiektów na mapie

Mapa w systemie wspomagania obserwacji astronomicznych jest mapą dwuwymiarową i pozycja obiektów położonych na niej jest określana w systemie współrzędnych kartezjańskich. Z tego powodu obliczone współrzędne horyzontalne nie wystarczą do poprawnego wyświetlenia obiektów na mapie, należy więc przeprowadzić dalszą konwersję polegającą na przekonwertowaniu współrzędnych horyzontalnych na współrzędne biegunowe, a następnie na współrzędne kartezjańskie. Aby przekonwertować współrzędne horyzontalne na współrzędne biegunowe wystarczy znać promień mapy, r, który jest promieniem wodzącym a azymut, A, można zatem uznać za amplitudę punktu w układzie współrzędnych biegunowych. Mając więc już współrzędne biegunowe można łatwo obliczyć potrzebne współrzędne kartezjańskie:

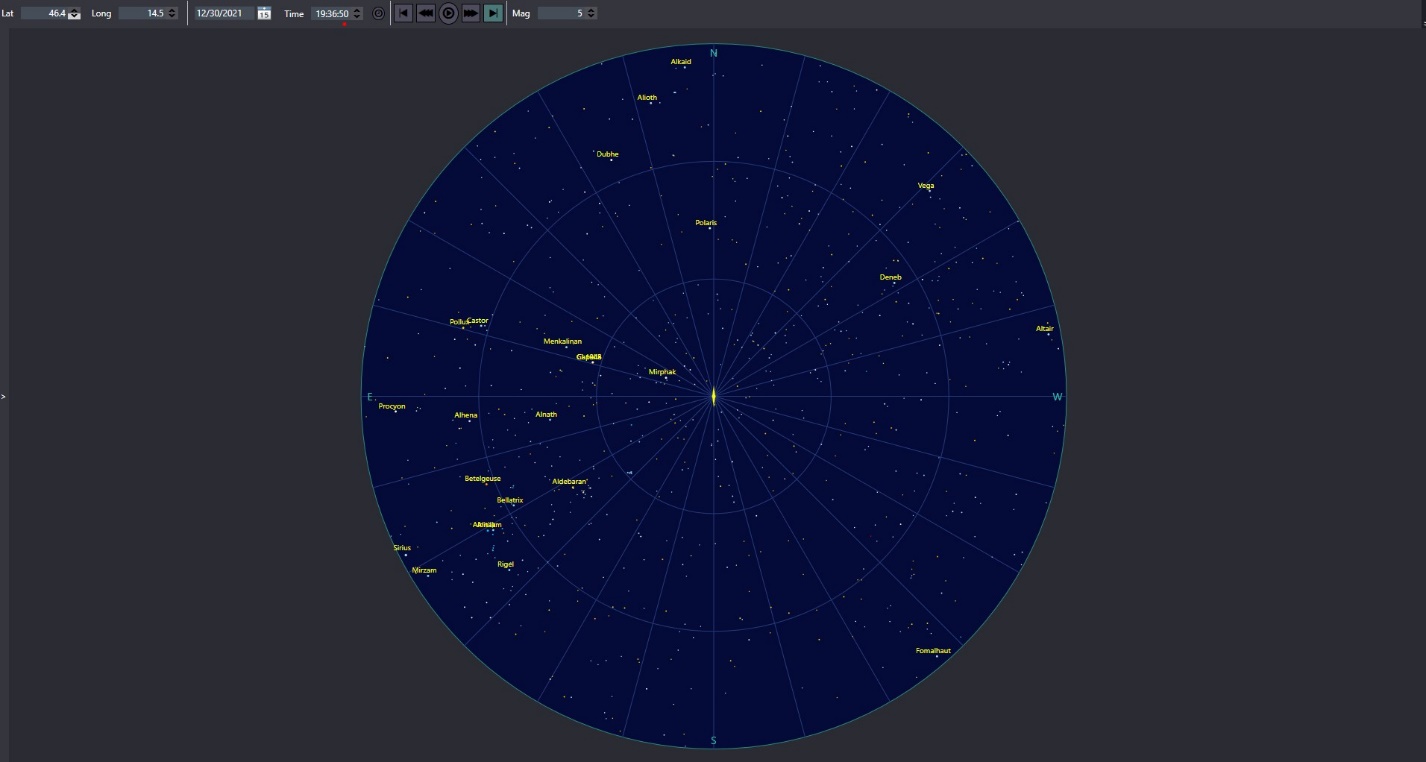
### 4.4.3 Wczytywanie obiektów astronomicznych

Obiektów wyświetlanych na sferze niebieskiej jest wiele dlatego algorytm wczytujący je został zaprojektowany tak aby jak najszybciej były one wczytywane. Prezentuje się on następująco:

|  |
| --- |
| 1. Wyczyścić czarną listę obiektów gdzie przechowywane są klucze obiektów, których baza danych ma nie wczytywać 2. Dla wczytanych wcześniej obiektów obliczyć pozycję na nowo, jeśli pozycja jest powyżej horyzontu to zaktualizować pozycję obiektu i dodać ją na czarną listę, w innym przypadku usunąć obiekt z kolekcji. 3. Wczytać obiekty z bazy danych, które dla podanych współrzędnych obserwacyjnych pojawiają się kiedykolwiek w ciągu roku oraz zignorować te, które są zapisane w czarnej liście, a następnie z tych wczytanych obiektów pobrać tylko te, które są na sferze niebieskiej dla podanych współrzędnych geograficznych o danym czasie 4. Obliczyć współrzędne horyzontalne dla pobranych obiektów 5. Dodać pobrane obiekty do kolekcji obiektów, które wyświetlane są na mapie |

## 4.5 Projekt interfejsu

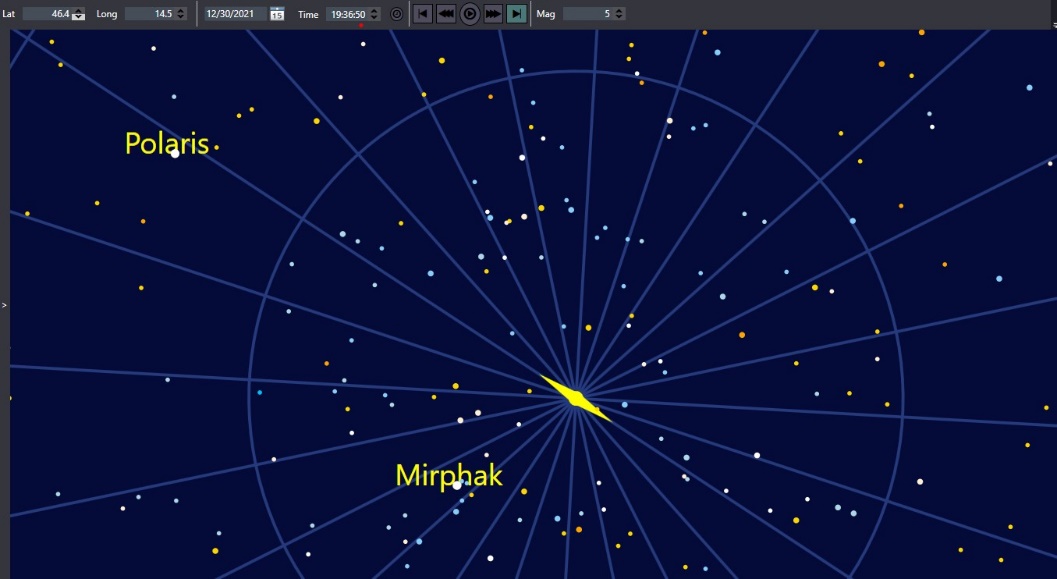
Interfejsem użytkownika jest tzw. przestrzeń, w której dochodzi do interakcji człowieka z maszyną. Jest to więc jeden z najważniejszych elementów projektu oprogramowania, który musi być zrozumiały oraz intuicyjny dla użytkownika, który będzie z niego korzystał. System wspomagania obserwacji astronomicznych zatem udostępnia interfejs, który posiada swoje elementy w odpowiednich, łatwo dostępnych miejscach.



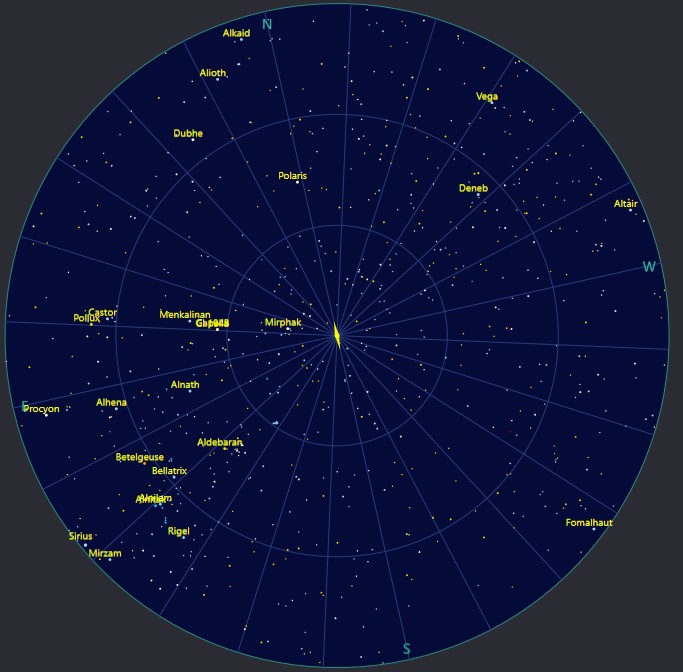
Rysunek 6 Główny widok interfejsu.

### 4.5.1 Mapa sfery niebieskiej

Najważniejszą częścią interfejsu jest mapa, na której wyświetlane są obiekty astronomiczne. Na mapie oznaczone są kierunki geograficzne oraz nałożona jest siatka, która ułatwia użytkownikowi rozeznać się w położeniu obszarów mapy, na które spogląda. Obok obiektów, które są jednymi z jaśniejszych na mapie wyświetlane są ich podstawowe nazwy. Użytkownik ma możliwość, za pomocą myszki, powiększyć mapę, obracać nią oraz przesuwać się po niej w danym kierunku.



Rysunek 7 Widok przybliżonej I obróconej mapy.



Rysunek 8 Pełny widok mapy sfery niebieskiej.

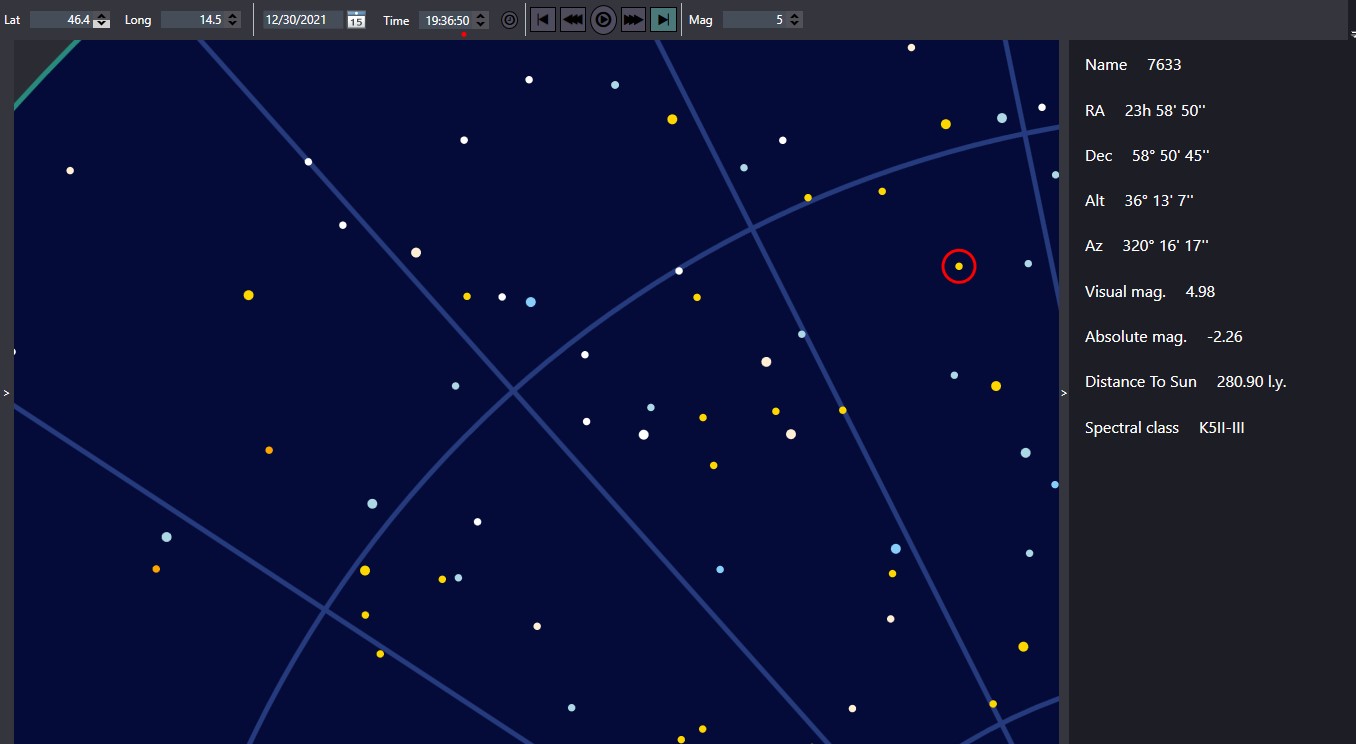
### 4.5.2 Obiekty astronomiczne

Na mapie wyświetlane są gwiazdy, które różnią się wyglądem na prawdziwej sferze niebieskiej, dlatego na mapie w systemie wspomagania obserwacji astronomicznych to też zostało uwzględnione. Każda gwiazda im jest jaśniejsza tym większy ma rozmiar na mapie oraz ma kolor odpowiedni do swojego typu widmowego. Po najechaniu myszką na jedną z gwiazd wyświetla się panel z podstawowymi informacjami o tej gwieździe, czyli: nazwa, współrzędne równikowe oraz współrzędne horyzontalne.



Rysunek 9 Grupa obiektów astronomicznych na mapie wraz z panelem informacyjnym tyczącym się jednej gwiazdy.

Każdy obiekt astronomiczny na mapie może zostać zaznaczony przez użytkownika poprzez kliknięcie go. W ten sposób wyświetlany jest panel boczny z dokładniejszymi informacjami na temat zaznaczonego obiektu, który w tym stanie jest zakreślony czerwonym okręgiem aby użytkownik wiedział, który został zaznaczony. Tymi informacjami są: nazwa, współrzędne równikowe oraz horyzontalne, obserwowalna wielkość gwiazdowa, absolutna wielkość gwiazdowa, dystans do Słońca oraz typ widmowy gwiazdy. Panel ten może być schowany ponownym kliknięciem na zaznaczoną gwiazdę.



Rysunek 10 Panel z informacjami na temat zaznaczonej gwiazdy.

### 4.5.3 Pasek narzędzi

W górnej części interfejsu został umieszczony pasek narzędzi, w którym użytkownik ma możliwość wprowadzić odpowiednie dane, od których zależy jaka część sfery niebieskiej zostanie ukazana na mapie. W pasku narzędzi do wprowadzenia lokalizacji umieszczone są dwie kontrolki odpowiadające długości i szerokości geograficznej, do których można wprowadzić tylko i wyłącznie wartości liczbowe w zakresie współrzędnych geograficznych dzięki czemu nie ma możliwości wprowadzenia błędnych danych. Również tyczy się to kontrolek, które odpowiadają za określenie czasu obserwacji. Dodany jest również przycisk wczytujący aktualną datę i czas użytkownika z systemu komputerowego, na którym użytkownik pracuje.

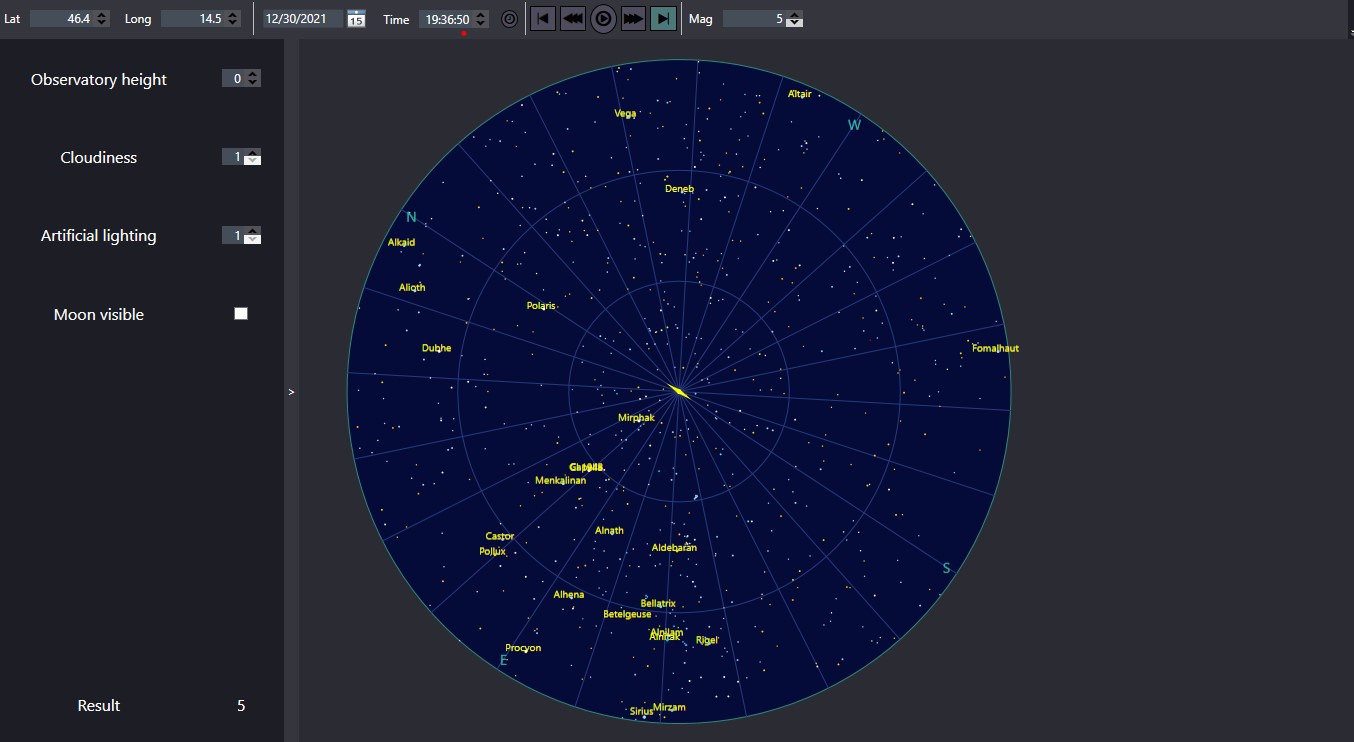


Rysunek 11 Pasek narzędzi.

W pasku narzędzi umieszczone są również przyciski kontrolujące upływ czasu. Można nimi więc uruchomić lub zatrzymać czas oraz określić kierunek, w którym będzie się poruszał czas i prędkość z jaką upływa czas (co jest zwizualizowane czerwoną kropką pod odpowiednią częścią wyświetlonego czasu na pasku narzędzi). Na końcu paska narzędzi jest również kontrolka do wprowadzenia minimalnej wielkości gwiazdowej obiektów, które mają zostać wyświetlone na mapie sfery niebieskiej.

### 4.5.4 Panel oceny warunków obserwacyjnych

W interfejsie systemu wspomagania obserwacji astronomicznych jest umieszczony panel, w którym użytkownik może wprowadzić dane cechy warunków pogodowych, a system wyprowadzi rezultat w postaci oceny liczbowej.



Rysunek 12 Panel oceny warunków pogodowych.

# 5. Testy systemu

W tym rozdziale zostanie przedstawiony proces manualnego testowania działania utworzonego systemu w celu weryfikacji czy funkcjonalności działają poprawnie. Każdy test będzie zapisany z opisanymi krokami, przy których zakładany jest wymagany rezultat.

## 5.1 Weryfikacja pobierania i wyświetlania obiektów na mapie

|  |  |  |
| --- | --- | --- |
| Krok | OPIS | OCZEKIWANE REZULTATY |
| 1 | Uruchomić system | System uruchamia się prawidłowo |
| 2 | Sprawdzić czy obiekty astronomiczne pojawiły się na mapie | Mapa wypełniona jest punktami reprezentującymi obiekty astronomiczne. |

Tabela 6 Kroki testujące pobieranie i wyświetlanie obiektów na mapie.

## 5.2 Weryfikacja interaktywności i sterowalności mapy

|  |  |  |
| --- | --- | --- |
| Krok | OPIS | OCZEKIWANE REZULTATY |
| 1 | Uruchomić system | System uruchamia się prawidłowo |
| 2 | Najechać myszką na jakikolwiek obszar mapy, wcisnąć i przytrzymać prawy przycisk myszy, następnie przesuwać kursor wokół środka mapy | Mapa obraca się zgodnie z kierunkiem kursora myszy |
| 3 | Najechać myszką na jakikolwiek obszar mapy, następnie obrócić kółko na myszce komputera w obie strony | Mapa powiększa się i zmniejsza względem obrotu kółka na myszy |
| 4 | Powiększyć mapę. Na powiększonej mapie wcisnąć i przytrzymać lewy przycisk myszy, następnie przesunąć mysz w górę, lewo, prawo i dół | Mapa przesuwa się zgodnie z ruchem myszy |

Tabela 7 Kroki testujące interaktywność i sterowalność mapy.

## 5.3 Weryfikacja wyświetlania informacji o danym obiekcie

|  |  |  |
| --- | --- | --- |
| Krok | OPIS | OCZEKIWANE REZULTATY |
| 1 | Uruchomić system | System uruchamia się prawidłowo |
| 2 | Najechać myszką na jakikolwiek obiekt astronomiczny umieszczony na mapie | Po chwili wyświetlany jest „tooltip” z informacjami o obiekcie |
| 3 | Na tym samym obiekcie wcisnąć lewy przycisk myszy | W prawej części widoku pojawia się panel z dokładniejszymi informacjami na temat obiektu |
| 4 | Na tym samym obiekcie wcisnąć lewy przycisk jeszcze raz | Panel po prawej stronie chowa się. |

Tabela 8 Kroki testujące wyświetlanie informacji o danym obiekcie.

## 5.4 Weryfikacja ustawiania lokalizacji i czasu obserwacji

|  |  |  |
| --- | --- | --- |
| Krok | OPIS | OCZEKIWANE REZULTATY |
| 1 | Uruchomić system | System uruchamia się prawidłowo |
| 2 | Zmienić współrzędne geograficzne obserwacji na inne niż domyślne | Mapa jest aktualizowana, obiekty zmieniają swoje położenie |
| 3 | Zmienić czas obserwacji na inny niż domyślny | Mapa jest aktualizowana, obiekty zmieniają swoje położenie |

Tabela 9 Kroki testujące ustawianie lokalizacji i czasu obserwacji.

## 5.5 Weryfikacja upływu i sterowania czasem

|  |  |  |
| --- | --- | --- |
| Krok | OPIS | OCZEKIWANE REZULTATY |
| 1 | Uruchomić system | System uruchamia się prawidłowo |
| 2 | Wcisnąć środkowy przycisk opowiadający za uruchomienie upływu czasu | Obiekty na mapie zaczynają zmieniać swoje położenie względem upływającego czasu. Czas i data obserwacji się zmieniają |
| 3 | Wcisnąć pięć razy pierwszy przycisk na prawo od przycisku środkowego odpowiadającego za przyspieszenie czasu | Czas po każdym kliknięciu upływa coraz szybciej aż do maksymalnego przyspieszenia czyli zwiększenie roku w dacie obserwacji co jedną sekundę. Obiekty na mapie zmieniają swoje położenie |
| 4 | Wcisnąć pięć razy pierwszy przycisk na lewo od przycisku środkowego odpowiadającego za zwolnienie czasu | Czas po każdym kliknięciu upływa coraz wolniej aż do minimalnego przyspieszenia czyli zwiększenie sekundy w czasie obserwacji co jedną sekundę. Obiekty na mapie zmieniają swoje położenie |
| 5 | Wcisnąć przycisk najbardziej na lewo odpowiadający za kierunek upływu czasu w przeszłość | Czas zaczyna upływać w kierunku przeszłości. Obiekty na mapie zmieniają swoje położenie |
| 6 | Wcisnąć przycisk najbardziej na prawo odpowiadający za kierunek upływu czasu w przyszłość | Czas zaczyna upływać w kierunku przyszłości. Obiekty na mapie zmieniają swoje położenie |
| 7 | Wcisnąć środkowy przycisk opowiadający za uruchomienie upływu czasu | Upływ czasu był już uruchomiony więc ponowne wciśnięcie przycisku powoduje zatrzymanie upływu czasu. |

Tabela 10 Kroki testujące upływ czasu i sterowanie nim.

## 5.6 Weryfikacja oceny warunków pogodowych

|  |  |  |
| --- | --- | --- |
| Krok | OPIS | OCZEKIWANE REZULTATY |
| 1 | Uruchomić system | System uruchamia się prawidłowo |
| 2 | Otworzyć panel oceny warunków pogodowych | Panel wyświetla się prawidłowo |
| 3 | Zmieniać wartości warunków pogodowych | Wprowadzane wartości nie wychodzą poza ich zakresy. Podsumowanie warunków pogodowych wyświetlane jest w postaci liczby ujemnej lub dodatniej. Wszystkie wartości muszą być zgodne z zdefiniowanymi kryteriami |

Tabela 11 Kroki testujące ocenę warunków pogodowych.

## 5.7 Podsumowanie testowania

Wykonanie wyżej opisanych testów potwierdza poprawność wykonania implementacji wszystkich wymaganych funkcjonalności. System więc działa prawidłowo i zostało to udowodnione dzięki zdefiniowanym przypadkom testowym. Świadczy to o tym, że testowanie jest ważną częścią projektowania oprogramowania w celu potwierdzenia działania wszystkich funkcji systemu lub wykryciu błędów, które można naprawić. Zmniejsza to ryzyko wystąpienia błędu w sytuacji kiedy to już użytkownik końcowy będzie korzystał z oprogramowania.

# 6. Wnioski i podsumowanie

Celem tej pracy było utworzenie systemu wspomagającego obserwacje astronomiczne. Przeanalizowany został rynek pod względem istniejących już rozwiązań, dzięki czemu można było określić co w systemie zaprojektować lepiej. Wykorzystano więc narzędzia ułatwiające pracę nad projektem, które przyspieszyły tworzenie go jak i zapewniły, że system będzie stabilny. Następnie przedstawiono zagadnienia astronomiczne, które trzeba było zrozumieć by móc taki system stworzyć. W ramach samego projektu systemu określono potrzebne funkcjonalności do implementacji, wybrano architekturę oraz utworzono bazę danych by ostatecznie zaimplementować wymagane algorytmy oraz interfejs użytkownika. Aby upewnić się, że projekt został utworzony poprawnie zostały zdefiniowane testy manualne na podstawie opisanych wcześniej funkcjonalności, które potwierdziły, że system jest zaprojektowany prawidłowo.

Architektura w projekcie został utworzona w taki sposób, że system może być dalej rozwijany dzięki czemu można wprowadzić potencjalne poprawki. Jedną z nich mogłoby być ulepszenie oceny warunków pogodowych poprzez automatyczne pobieranie informacji o pogodzie z Internetu. Drugą poprawką mogłoby być zoptymalizowanie algorytmu wypełniającego mapę obiektami astronomicznymi.

W ostateczności, w porównaniu do przeanalizowanych wcześniej systemów dostępnych na rynku, powstał system posiadający mapę, której interaktywność lepiej działa, oraz dwie nowe funkcjonalności, czyli ocena warunków obserwacyjnych oraz symulacja upływu czasu. Dzięki temu narzędziu obserwacje astronomiczne będą ułatwione, a korzystanie z niego jest przyjemne dla użytkownika.

# **Bibliografia**

[1] Hannu K., Pekka K., Heikki O., Markku P., Karl Johan D. 2017. Astronomia ogólna

[2] Peter Duffet-Smith, Jonathan Zwart. Practical Astronomy with your Calculator or Spreadsheet, fourth edition

[3] Ian Sommerville. Inżynieria oprogramowania

[4] Gamma, Helm, Johnson. Vlissides. Wzorce projektowe

[5] Joseph Albahari, Ben Albahari. C# 6.0 w pigułce

[6] Syme, Granicz, Cisternino. F# 4.0 dla zaawansowanych

[7] MacDonald Matthew. Pro WPF 4.5 in C#

[8] <https://www.astronexus.com/hyg>