

**UNIWERSYTET RZESZOWSKI**

**Kolegium Nauk**

Piotr Stec

Nr albumu: 106475

Kierunek  
Informatyka

**Tytuł (temat) pracy dyplomowej**

**Aplikacja webowa przeznaczona do pozyskiwania podstawowych informacji o gwiazdach**

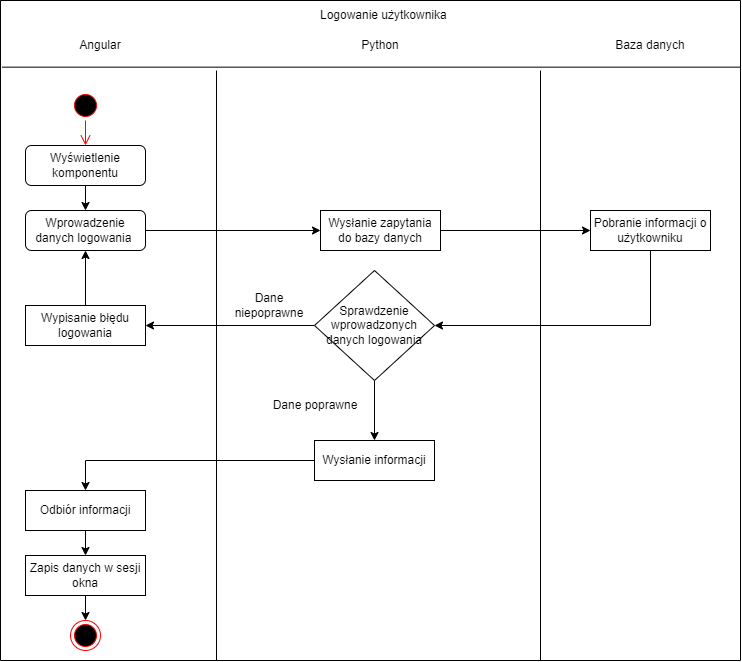
Praca licencjacka/magisterska/inżynierska

Praca wykonana pod kierunkiem

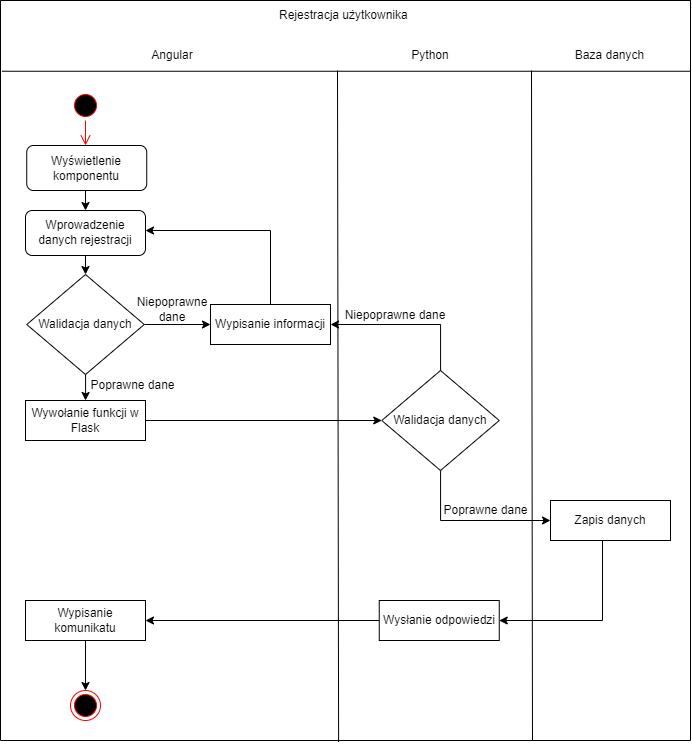
….....……………………………………..

Rzeszów, data

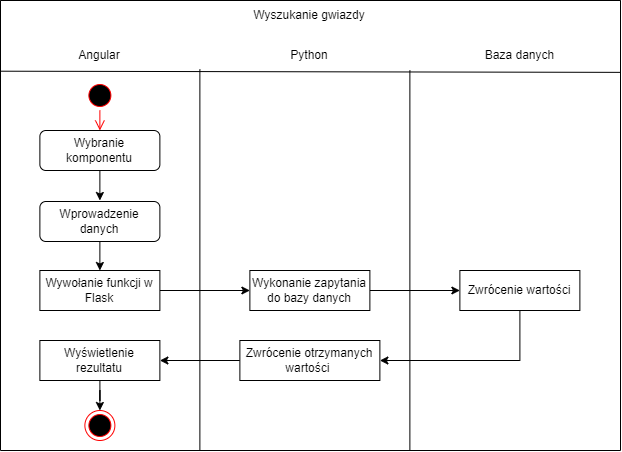
1. **Wstęp**W dzisiejszym świecie, stawiamy czoła coraz to nowym i nieznanym rzeczom. Potrzeba odkrywania jak udoskonalania stanowi meritum dalszego rozwoju człowieka. Dzięki staraniom wielu ludzi, potrafimy zebraną wiedzą zapisać i przechować dla przyszłych odkrywców. Największą liczbę nowych i zaskakujących odkryć można zaobserwować podczas poznawania kosmosu. Od początku istnienia ludzie patrzyli w górę i widzieli bezkres świata, tysiące małych punktów, które bez względu na czas, były, są i będą. Gwiazdy zachwycały oraz inspirowały wielu znakomitych umysłów. Ludzie nauczyli się dokładnie opisywać i przekazywać tą wiedzą o tym co widzą ponad swoją głową. Dzisiaj kiedy każdy ma swobodny dostęp do internetu, a wyszukanie informacji jest proste i szybkie. Zdobycie wiedzy o otaczającym nas świecie nie stanowi wyzwania. W dzisiejszych czasach możliwość odkrycia nowej i nie zbadanej rzeczy jest prostsza niż kiedykolwiek.  
   1. **Cel i zakres pracy**Celem projektu było stworzenie aplikacji internetowej, która pozwoli osobom zainteresowanym podziwianiem nocnego nieba, rozwijać swoje hobby jak i da im możliwość wyróżnienia się w konstruowaniu wiedzy o tych obiektach.   
      Aplikacja pozwala na prosty i intuicyjny dostęp do informacji o gwiazdach i gwiazdozbiorach. Czynnikiem szczególnym, jest możliwość dodania gwiazdy do istniejącej bazy danych. Ta opcja jest przewidziana dla każdego zarejestrowanego użytkownika. Do kontroli nowo utworzonych elementów przewidziana jest rola administratora, który sprawowałby nadzór nad rozwojem treści dodawanych jak i już istniejących.
   2. **Motywacja**Jako osoba lubiąca poszerzać swoją wiedzę z dziedziny gwiazd, podczas szukania informacji, często napotykaną sytuacją był problem z zrozumieniem treści opisującej dany obiekt astronomiczny. Ilość danych jak sposób ich przedstawienia, w większości przypadków wskazuje na to, że odbiorcom powinien być naukowiec posiadający szeroką wiedzę astronomiczną. Innym nie rzadko napotykanym problemem było odnalezienie informacji o gwieździe znając jedynie jej przynależność do gwiazdozbioru. Na wielu witrynach internetowych dotyczących gwiazd jest możliwe znalezienie bardzo dokładnych informacji lecz pod warunkiem dobrej znajomości szukanego obiektu. Mimo znalezienia gwiazdy, której szukamy duża część przedstawianej wiedzy wymaga bardzo dobrej orientacji na temat terminologi stosowanej przez astronomów, ten fakt skłonił mnie do stworzenia aplikacji, która nie zalewa użytkownika cyframi lub tekstem lecz przedstawia najważniejsze dane w prosty i wygodny sposób.
2. **Technologie wykorzystane w projekcie**
   1. **MariaDB** Jest to system przeznaczony do zarządzania relacyjną bazą danych. Został on zaprojektowany przez byłych pracowników MySQLAB. Twórcy za główny cel wyznaczyli sobie stworzenie w pełni wolnego oprogramowania udostępnianego na licencji GPL. MariaDB jest przeznaczona do wysokiej kompatybilności z MySQL oraz jest stale rozwijana o nowe funkcjonalności i usprawnienia.
   2. **Python**Jest to język programistyczny wysokiego poziomu umożliwiający mnogą liczbę możliwości rozwiązania jednego problemu. Posiada on bardzo rozbudowaną i aktywną społeczność udostępniającą darmowe biblioteki do szerokiego zastosowania w informatyce. W projekcie w głównej mierze biorą udział dwie biblioteki, Flask oraz SQLAlchemy. Pomocniczymi rozszerzeniami są: hashlib, Abstract Base Classes, pandas.
   3. **Angular**Jest to język programistyczny wysokiego poziomu umożliwiający mnogą liczbę możliwości rozwiązania jednego problemu. Posiada on bardzo rozbudowaną i aktywną społeczność udostępniającą darmowe biblioteki do szerokiego zastosowania w informatyce. W projekcie w głównej mierze biorą udział dwie biblioteki, Flask oraz SQLAlchemy. Pomocniczymi rozszerzeniami są: hashlib, Abstract Base Classes, pandas.
   4. **Bootstrap 4**Darmowa, udostępniona na licencji Apache License 2.0, nakładka stworzona do tworzenia responsywnych i dynamicznych aplikacji internetowych przeznaczonych zarówno na komputery jak i telefony.
3. **Diagramy aktywności**
   1. **Diagram logowania użytkownika**Użytkownik rozpoczyna przegląd aplikacji na stronie głównej, aby się zalogować powinien przejść do komponentu logowanie wciskając napis „Logowanie” znajdujący się na pasku zadań w górnej części witryny. Po tej akcji zostaną wyświetlone dwa pola: nazwa użytkownika oraz hasło. W celu zalogowania do aplikacji należy wprowadzić poprawne dane, które widnieją w bazie danych w przypadku nie posiadania konta w aplikacji użytkownik musi przejść proces rejestracji. Pola są sprawdzane w przypadku gdyby użytkownik omyłkowo potwierdzi puste dane, zostanie mu zwrócona uwaga o braku danych. Dodatkowym zabezpieczeniem jest objęte pole zawierające hasło. Aplikacja wymaga, aby wpisywany tekst zwierał co najmniej z 8 znaków w przypadku błędnego wpisania tekstu zostanie pokazany komunikat o za krótkim ciągu znaków. Po zatwierdzeniu danych wywołana zostanie funkcja w aplikacji backendowej, sprawdzi ona poprawność wpisanego hasła dla użytkownika o podanej nazwie, jeśli warunek zostanie spełniony zostanie wysłana odpowiedź zawierająca informacje o poprawnym zalogowaniu użytkownika. Aplikacja będzie przechowywała dane logowania do momentu zamknięcia okna bądź wylogowania użytkownika.   
      Program w celu porównania hasła otrzymanego od użytkownika a zapisanego w bazie danych, najpierw tworzy hash na podstawie otrzymanego tekstu, a następnie porównuje go z wartością przechowywaną w serwerze MariaDB. Wszystkie hasła zapisane w bazie danych są zakodowane algorytmem „PBKDF2\_HMAC”. Do tworzenia ciągu znaków wymagane jest podanie czterech elementów: funkcji hashu, ziarna generatora, liczby iteracji oraz tekst, który chcemy zakodować. Funkcja hashu to „SHA256”, przekłada się ona na możliwość stworzenia ciągu znaków o maksymalnej długości 256 bitów. Generator wymaga ziarno, które będzie fundamentem budowy hashu. Finalnym elementem jest liczba iteracji tworząca poziom komplikacji hasła.  
      Taka metodyka jest często stosowna na prstych platformach internetowych, zapewnia ona dodatkową ochronę w przypadku gdyby dane z bazy zostały wykradzione. Dla osoby nie znającej czterech elementów składających się na generowany hash jest to losowy ciąg znaków.

Rysunek 1: Diagram logowania

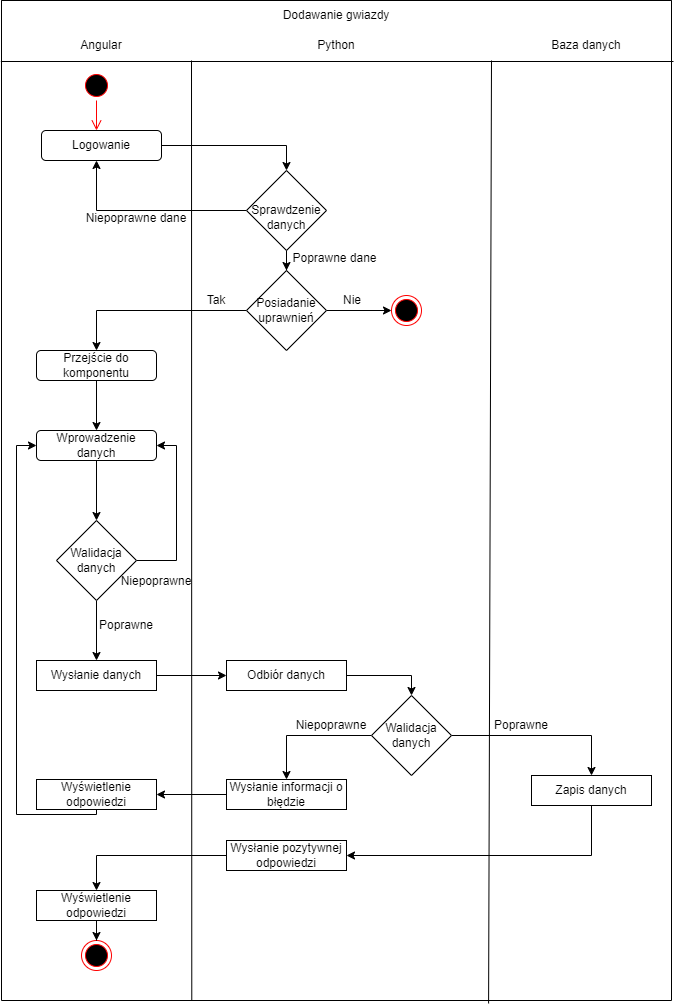
* 1. **Diagram rejestracji użytkownika**Użytkownik odwiedzający pierwszy raz aplikację i chcący uczestniczyć w jej rozwoju może się w niej zarejestrować, aby tego dokonać z paska zadań na stronie głównej musi wybrać napis „Rejestracja”. Zostanie on wtedy przekierowany do komponentu z formularzem tam   
     użytkownik jest zobowiązany do uzupełnienia wszystkich pól tekstowych. Wpisane dane będą wykorzystywane podczas dalszego korzystania z aplikacji. Dla kilku danych zostały przewidziane szczególne warunki, które muszą zostać spełnione.  
     Dla pola mail wpisywana wartość musi odzwierciedlać rzeczywisty format maila. Oznacza to, że wpisywana wartość posiada znak ‘@’, nazwę usługodawcy oraz domenę. Dodatkowo pole to jest wymagane jako unikatowe w bazie danych. Wartość maila jest sprawdzana dwukrotnie raz w części frontendowej oraz drugi podczas działania metody w Flask-u.  
     Polem obłożonym dodatkowym warunkiem jest nazwa użytkownika. Wartość ta musi być unikatowa dla aplikacji. Jest ona sprawdzana dwukrotnie, za pomocą warunku w backendize oraz bezpośrednio w bazie danych.  
     Dla hasła przewidziany jest warunek sprawdzający długość wpisanego tekstu, minimalna wartość to 8 znaków. Wartość ta musi zostać powtórzona dla pola „Powtórz hasło” w przypadku błędnego zapisania danych zostanie zwrócony komunikat informujący o popełnionym błędzie.   
     Dane istotne do wyróżnienia użytkownika to imię i nazwisko będą one wykorzystywane w późniejszej części aplikacji.

Rysunek 2: Diagram rejestracji

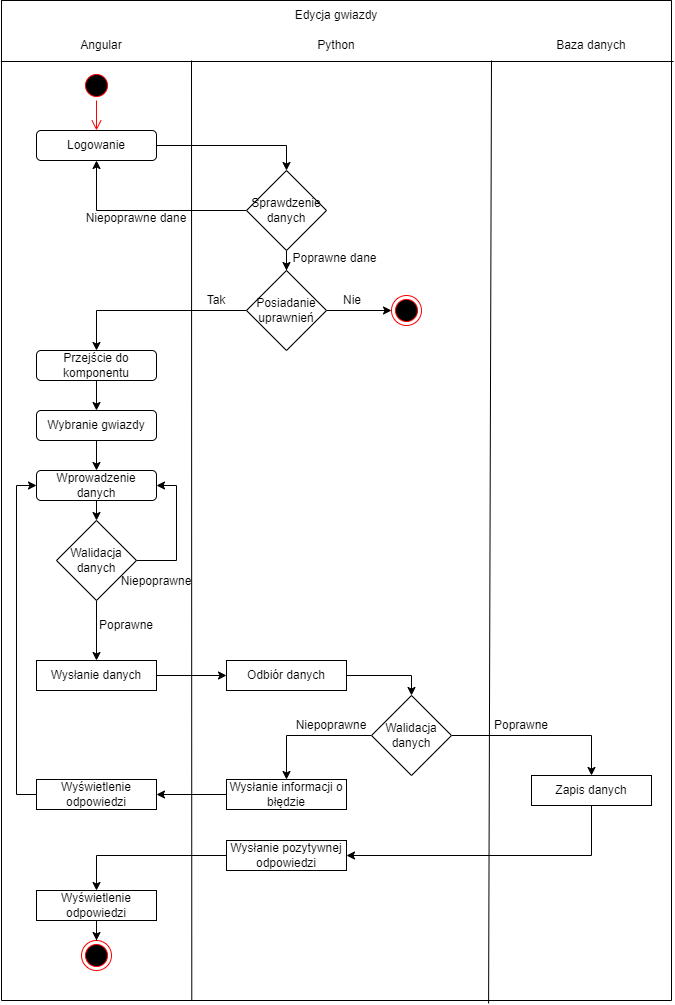
* 1. **Diagram wyszukiwania gwiazdy**Jedną z dostępnych opcji w aplikacji jest wyszukanie informacji o gwieździe. Odbywa się ono poprzez komponent „Szukaj gwiazdę”. Użytkownikowi po wybraniu opcji pokaże się pole tekstowe, w które może wprowadzić nazwę szukanej gwiazdy. Po potwierdzeniu wpisanej nazwy zostanie wywołana funkcja w aplikacji backendowej. Zostanie do niej przekazana treść wpisana w pole wyszukiwania. Aplikacja wykona zapytanie do bazy danych poprzez SQLAlchemy o wpisaną frazę. Wyszukiwanie następuje po kolumnie z nazwą gwiazdy metoda zwraca wynik zawierający co najmniej otrzymany tekst oznacza to, że użytkownik może pamiętać część nazwy gwiazdy a aplikacja zwróci wartości podobne. Rezultat zapytania jest ograniczony do 25 obiektów, jest to zabieg zastosowany w celach zachowania czytelności jak i redukcji czasu wyświetlenia odpowiedzi. Finalnie rezultatem kliknięcia przycisku jest wyświetlenie w tabeli listy gwiazd.

Rysunek 3: Diagram wyszukiwania gwiazdy

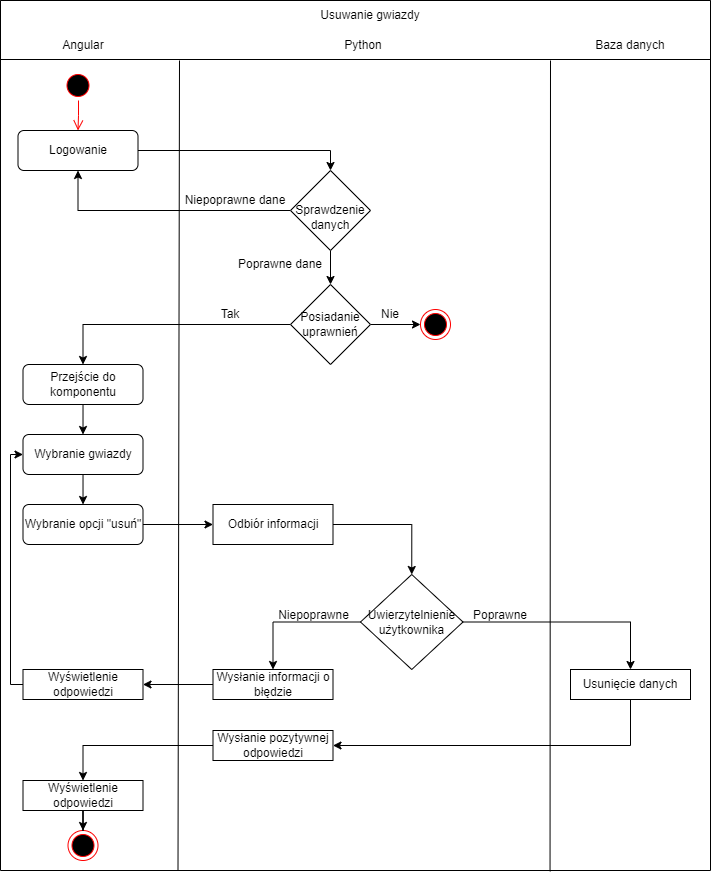
* 1. **Diagram dodawania nowej gwiazdy**Dodawanie gwiazdy jest dostępne jedynie dla zalogowanego użytkownika. Aplikacja po otrzymaniu informacji o użytkowniku, zmieni układ paska zadań, w którym pokaże się nowa opcja, „Dodaj gwiazdę”. Użytkownik po kliknięciu w link zostanie przeniesiony do formularza. W tym miejscu użytkownik może wprowadzić dane nowej gwiazdy. Wpisywane informacje są testowane za pomocą języka TypeScript. Pierwszym sprawdzanym warunkiem jest wartość null   
     w wymaganych pola. Następnymi w kolejności elemenatmi są wartości liczbowe, muszą one spełniać warunki ograniczenia dla odpowiednich parametrów. Wartości rektascencji zawierają się w przedziale od 0 do 24, dla deklinacji jest to od -90 do 90 stopni. Część ułamkowe tych dwóch współrzędnych muszą być w przedziale od 0 do 60. Gdy zostanie naruszona jedna z reguł, użytkownik dostanie informację o popełnionym błędzie. Po spełnieniu wszystkich warunków wprowadzone dane są wysłane do funkcji w Flask. Tam sprawdzana jest rola użytkownika w aplikacji jak też jest przeprowadzana ponowna walidacja danych liczbowych i tekstowych. Częścią informacji, która jest sprawdzana w bazie danych to nazwa gwiazdozbioru musi ona się pokrywać z tymi zapisanymi w bazie danych. Po pozytywnym przejściu testów zostaje wywołane zapytanie do bazy danych o dodanie nowego obiektu. Po zapisie danych przesyłana jest informacja o poprawnie dodanym obiekcie, która z kolei jest przekazana i wyświetlona użytkownikowi. Każda nowo dodana gwiazda, przechodzi w późniejszym czasie dodatkowe sprawdzenie przez administratora. Jest to niezbędny krok w trakcie zapisu danych udostępnianych do wglądu użytkowników. Nowo dodana gwiazda nie jest możliwa do wyszukania w aplikacji bez posiadania praw administratora.

Rysunek 4: Diagram dodawania nowej gwiazdy

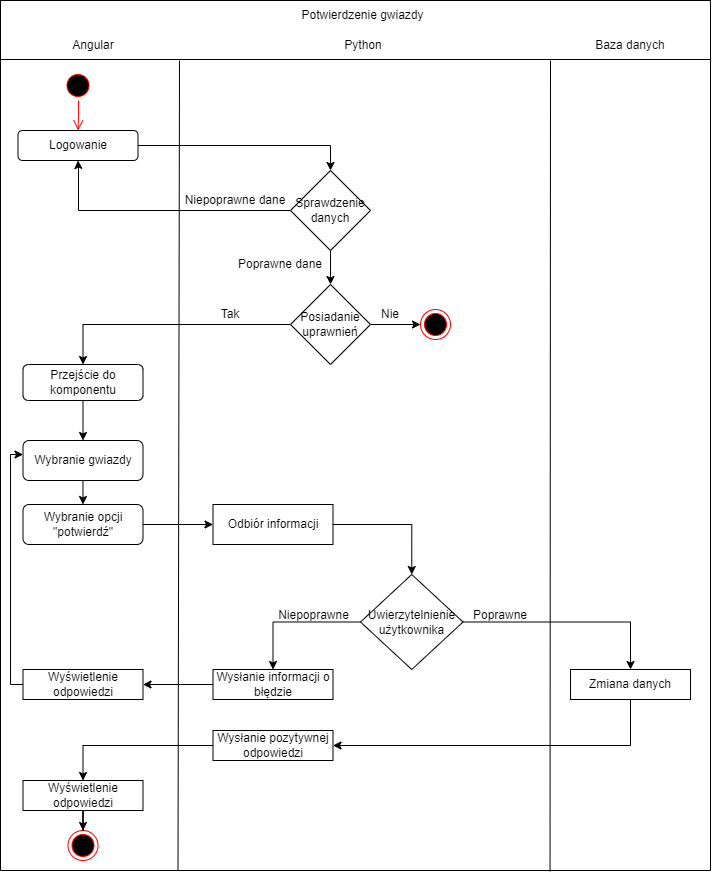
* 1. **Diagram edycji gwiazdy**Edycja gwiazdy jest dostępna jedynie dla konta administratora po zalogowaniu. Gdy użytkownikowi uda się zalogować do aplikacji w lekko zmienionym pasku zadań pokaże się opcja pokazania wszystkich gwiazd znajdujących się w bazie. Tam też administrator ma możliwość wyszukania gwiazdy i kolejno jej edycję. Po wybraniu tej opcji pokaże się formularz z uzupełnionymi danymi z bazy. Po zmianie lub dodaniu wartości do pól administrator potwierdza formularz, dane zostają . Dane zostają wysłane do funkcji w Flask, która z kolei sprawdza otrzymane wartości jak i to czy osoba wysyłająca formularz ma uprawnienia administratora.

Rysunek 5: Diagram edycji gwiazdy

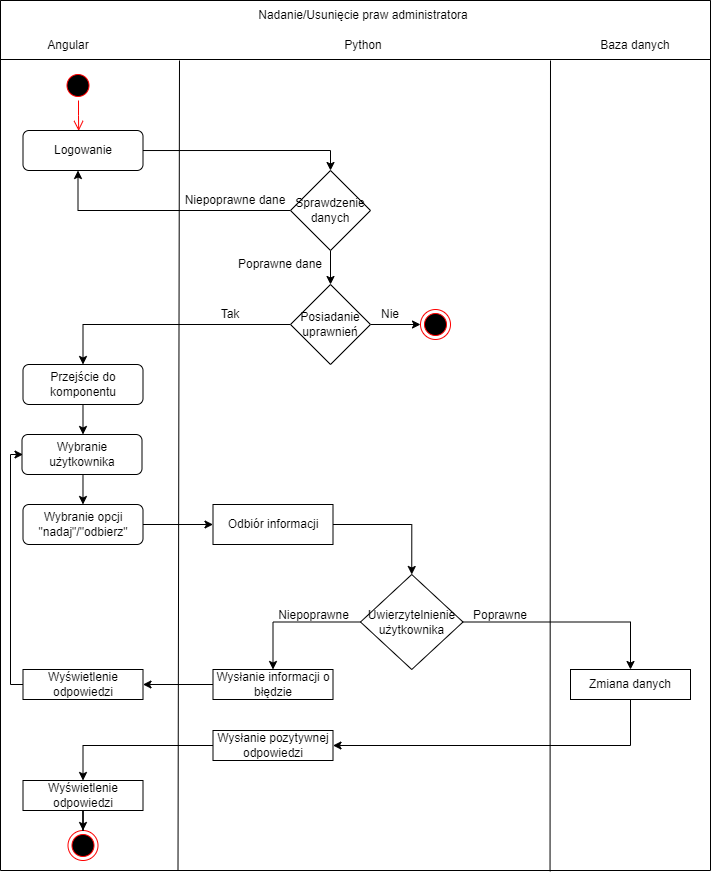
* 1. **Diagram usuwania gwiazdy**Usuwanie gwiazdy jest dostępne dla zalogowanego konta administratora. Opcja ta jest dostępna w komponencie wyświetlającym listę gwiazd w bazie danych. Usuwanie działa dla dwóch typów danych potwierdzonych i niepotwierdzonych. Aby usunąć gwiazdę użytkownik wyszukuje z listy obiekt a następnie wciska przycisk „Usuń”. Gdy nie można zlokalizować gwiazdy w tabeli można ją wyszukać poprzez wpisanie nazwy w polu wyszukiwania. Po wykonaniu akcji przez użytkownika zostaje wywołana funkcja w Flask. Frontend przekazuje niezbędne informacje o obiekcie do usunięcia. Backend odbiera dane, a następnie sprawdza uprawnienia użytkownika, który chcę wykonać operację. W przypadku niepoprawnej weryfikacji zostanie zwrócona informacja o braku uprawnień dla danego konta. Jeśli test zakończy się pozytywnym potwierdzeniem danych konta zostanie wysłane zapytanie do bazy danych z instrukcją usunięcia obiektu o wskazanych danych. Po wykonaniu instrukcji zostanie zwrócona informacja do backendu, która to przekaże ją jako wiadomość dla aplikacji frontendowej. Na końcu lista zostanie zaktualizowana a użytkownik zobaczy informację o usunięciu gwiazdy.

Rysunek 6: Diagram usuwania gwiazdy

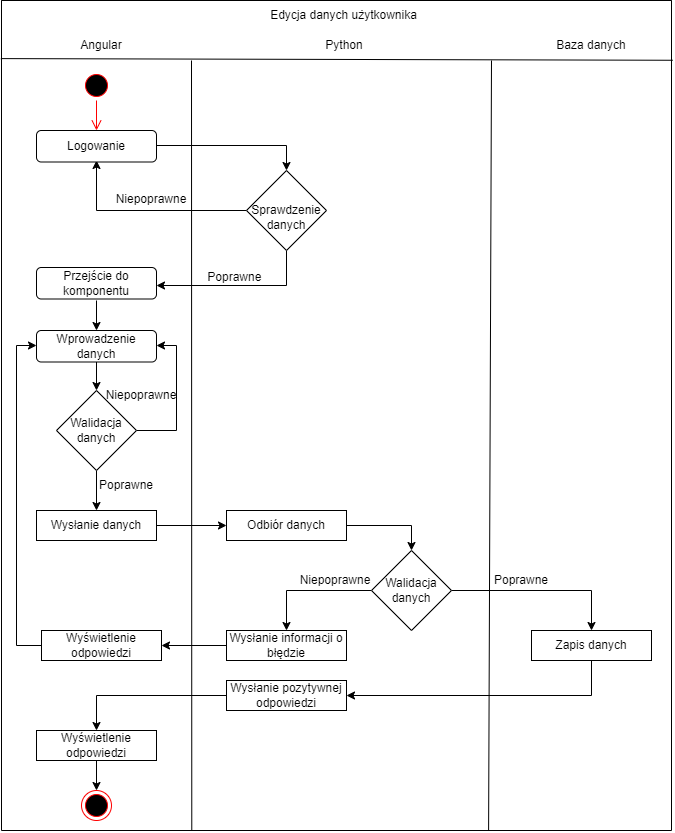
* 1. **Diagram potwierdzenia gwiazdy**Opcja potwierdzenia istnienia gwiazdy jest dostępna dla konta administratora. W celu dokonania tej czynności administrator musi przejść do zakładki „Lista formularzy”, i wybrać opcję „Niepotwierdzone gwiazdy”. W tym segmencie programu pokaże się lista formularzy,które wymagają potwierdzenia, gdy lista jest pusta zostanie zwrócona informacja o braku danych. W każdym rekordzie niepotwierdzonego formularza w tabeli istnieje przycisk o nazwie „Potwierdź”. Wybranie tej akcji spowoduje wywołanie funkcji w Flask. Przyjmuje ona informacje, o obiekcie do edycji oraz informacje o użytkowniku wysyłającym dane. Program sprawdza uprawnienia osoby wywołującej funkcję. Po potwierdzeniu następuje wywołanie funkcji w SQLAlchemy, która to edytuję kolumnę w bazie, zmieniając tym samym status gwizdy na potwierdzony. Po zakończeniu operacji zostaje zwrócona odpowiedź do frontendu, a tam wyświetlenie pozytywnej wiadomości.

Rysunek 7: Diagram potwierdzenia gwiazdy

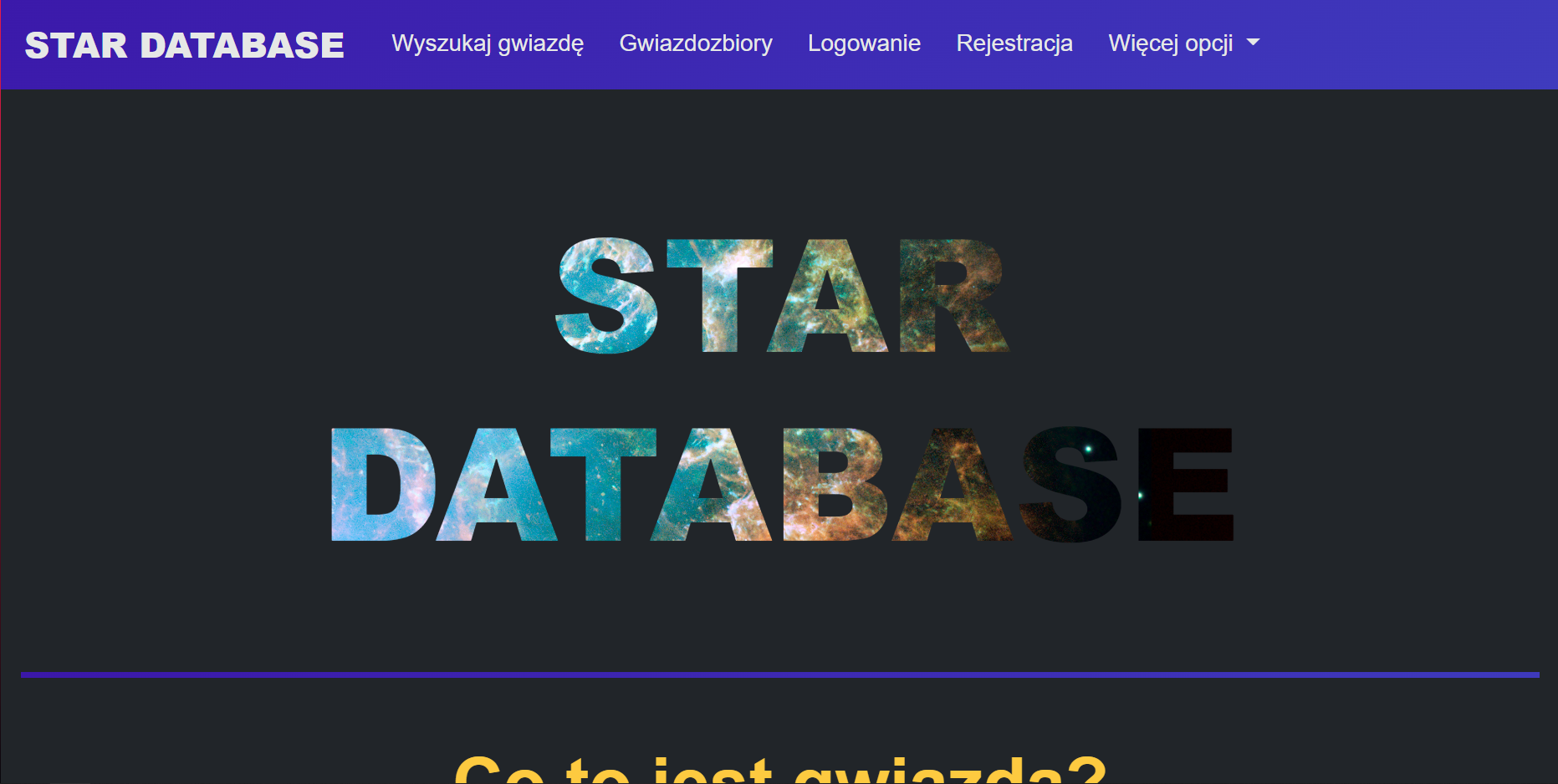
* 1. **Diagram nadania/odebrania praw administratora**W aplikacji przewidziane są dwie role użytkownik oraz administrator. Każda osoba zarejestrowana w aplikacji może zostać administratorem. Pełnomocnictwo nadania wyższych praw ma jedynie administrator, aby zmienić rolę użytkownika należy zalogować się do aplikacji. Po tej czynności w pasku zadań dostaniemy opcję przejścia do „Listy formularzy”, a tam wybrania zakładki z użytkownikami. W tabeli z listą aktualnie zarejestrowanych użytkowników zostają pokazane niezbędne dane do weryfikacji osoby takie jak imię, nazwisko oraz liczba odkrytych gwiazd. Na końcu każdego wiersza w tabeli znajduje się przycisk „Nadaj”. Po wybraniu osoby administrator kilka przycisk po czym zostaje wywołana funkcja w Flask. Aplikacja backendowa po otrzymaniu danych sprawdza czy osoba wysyłająca informacje posiada uprawnienia do ich edycji. Po pomyślnym przejściu warunku, metoda w SQLAlchemy zmienia dane w bazie czym jednocześnie nadaje nowe uprawnienia danemu użytkownikowi. Po poprawnym wykonaniu akcji zostaje zwrócona wiadomość na frontend, informująca o poprawnym zakończeniu edycji.  
     Operacja odbierania uprawnień administratora jest bardzo podobna. W miejscu przycisku z nadaniem praw pojawia się opcja usunięcia uprawnień. Tak jak w uprzedniej operacji następują po sobie procedury weryfikacji, edycji rekordu w bazie danych i wysłanie wiadomości o poprawnym przebiegu operacji.

Rysunek 8: Diagram nadania/odebrania praw administratora

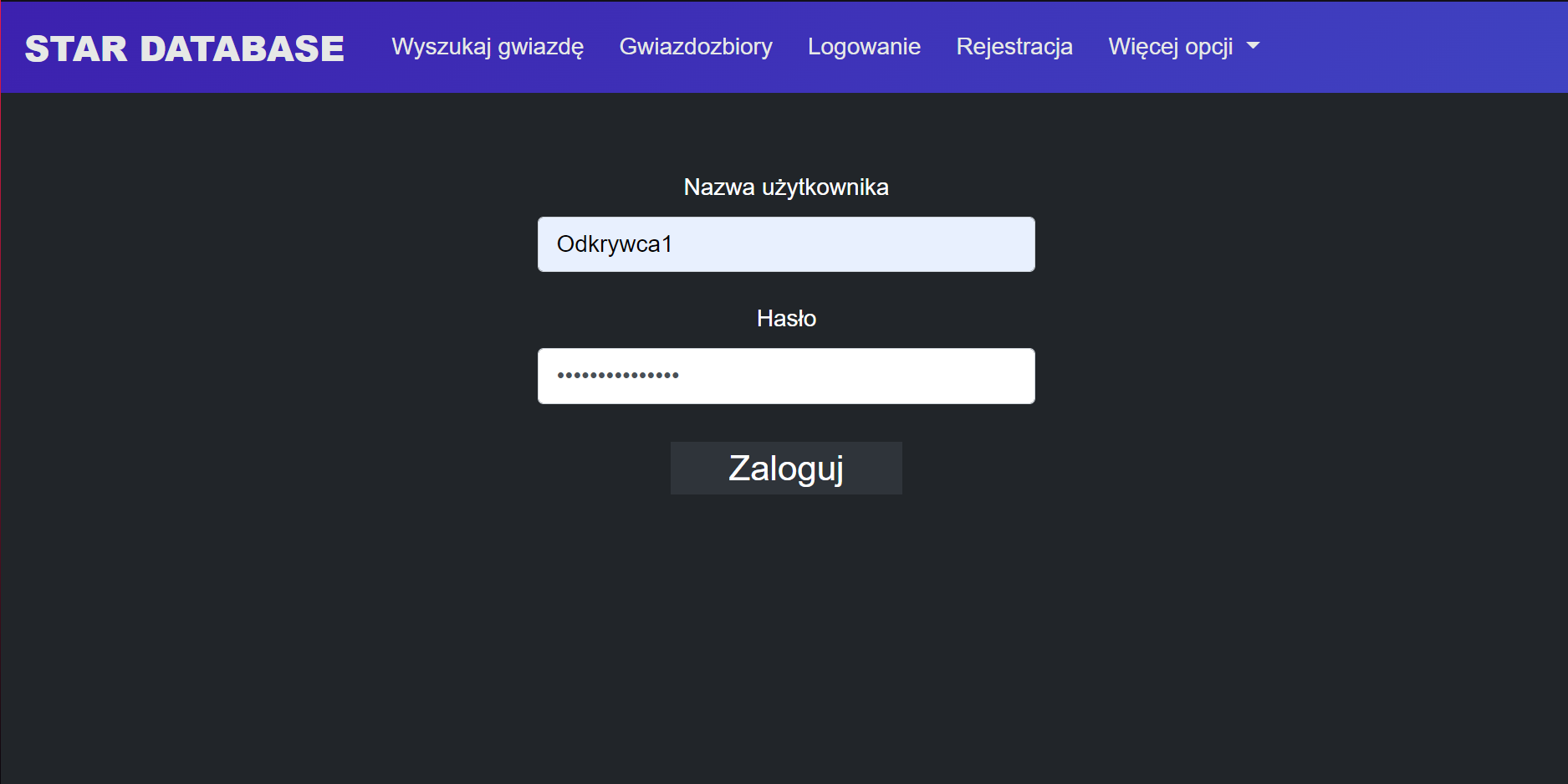
* 1. **Diagram edycji danych użytkownika**Każdy użytkownik posiada możliwość edycji swoich danych logowania. Aby wykonać operację zmiany dana osoba musi być zalogowana. Po tej czynności w pasku zadań zostaje pokazana opcja do poglądu informacji o swoim koncie. W zakładce „Profil” są pokazane dane o aktualnie zalogowanym użytkowniku oraz przycisk „Edycja danych logowania”, po kliknięciu na tę opcję zostaje wyświetlony formularz do edycji danych. Użytkownik po wprowadzeniu danych potwierdza formularz czym wywołuje funkcję w Flask. Tam sprawdzane są nowo wprowadzone dane jak i uwierzytelnienie osoby wysyłającej informacje. Po pozytywnym przejściu warunków następuje zmiana danych w bazie, a następnie zwrócona informacja o pozytywnym przebiegu akcji. W przypadku gdy nastąpi błąd bądź dane nie przejdą testów zostanie zwrócona informacja o niepoprawnym wykonaniu operacji. Elementami poddanymi warunkom są: unikatowość nazwy, długość hasła, zgodność zalogowanego użytkownika oraz tego istniejącego w bazie.

Rysunek 9: Diagram edycji danych użytkownika

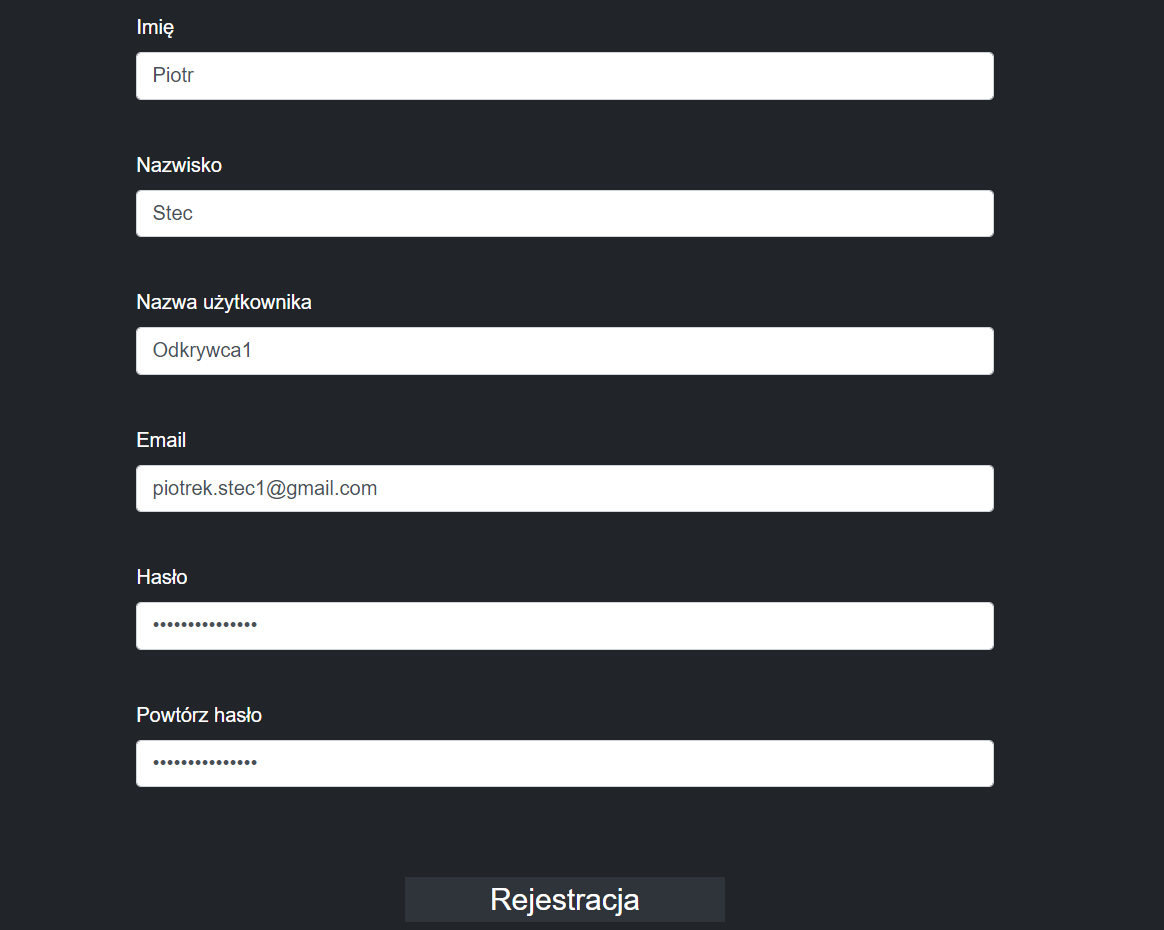
1. **Prezentacja aplikacji**
   1. **Główna strona**Użytkownikowi po wejściu do aplikacji ukazuje się napis z tytułem projektu oraz pasek zadań, znajdujący się w górnej części okna. Pod wizytówką aplikacji znajdują się podstawowe informacje o gwieździe jak i jednostkach astronomicznych wykorzystywanych w aplikacji. Są to podstawowe informacje szeroko dostępne w internecie a są one umieszczone w aplikacji po to aby pozwolić na uniknięcia nieporozumień związanych z różnorodnością przyjętych jednostek opisujących gwiazdy. Informacje te są w stanie zapewnić użytkownikowi wystarczającą wiedzę pomagającą w łatwym zrozumieniu wielkości i skali wykorzystywanych w astronomii.  
      Dla użytkownika niezalogowanego w aplikacji pasek zadań zwiera podstawowe opcje przewidziane w aplikacji. Są to możliwość rejestracji, zalogowania się, wyszukania gwiazdy, wyświetlenia informacji na temat gwiazdozbiorów, pokazanie ciekawostek oraz pokazanie danych o autorze projektu.

Rysunek 10: Główny widok aplikacji

* 1. **Logowanie**Użytkownik posiadający konto w aplikacji może się do niej zalogować wprowadzając swoje dane w komponencie do logowania. Rezultatem poprawnego zalogowania będzie dostęp do większej ilości opcji w aplikacji jest to możliwość dodania gwiazdy do bazy danych oraz dostęp do swojego profilu.   
     Jeśli użytkownik ma nadane prawa administratora to posiada on dostęp do edycji bazy danych, potwierdzenia formularza z informacjami o nowej gwieździe, opcji usuwania gwiazdy jak i modyfikacji jej danych. Dodatkową opcją jest możliwość nadania praw administratora innym użytkownikom.

Rysunek 11: Widok panelu logowania

* 1. **Rejestracja**Rejestracja w aplikacji odbywa się poprzez komponent stworzonym w Angularze. Użytkownik przechodząc do zakładki „Rejestracja” dostaje pusty formularz, aby utworzyć konto w aplikacji musi on wprowadzić dane do wszystkich widniejących pól, są to: imię, nazwisko, mail, hasło, nazwa użytkownika. Parametry takie jak mail oraz nazwa użytkownika muszą być unikatowe. Jest to stworzone w celu zapobiegania błędnej weryfikacji użytkownika w trakcie działania aplikacji.

Rysunek 12: Widok panelu rejestracji

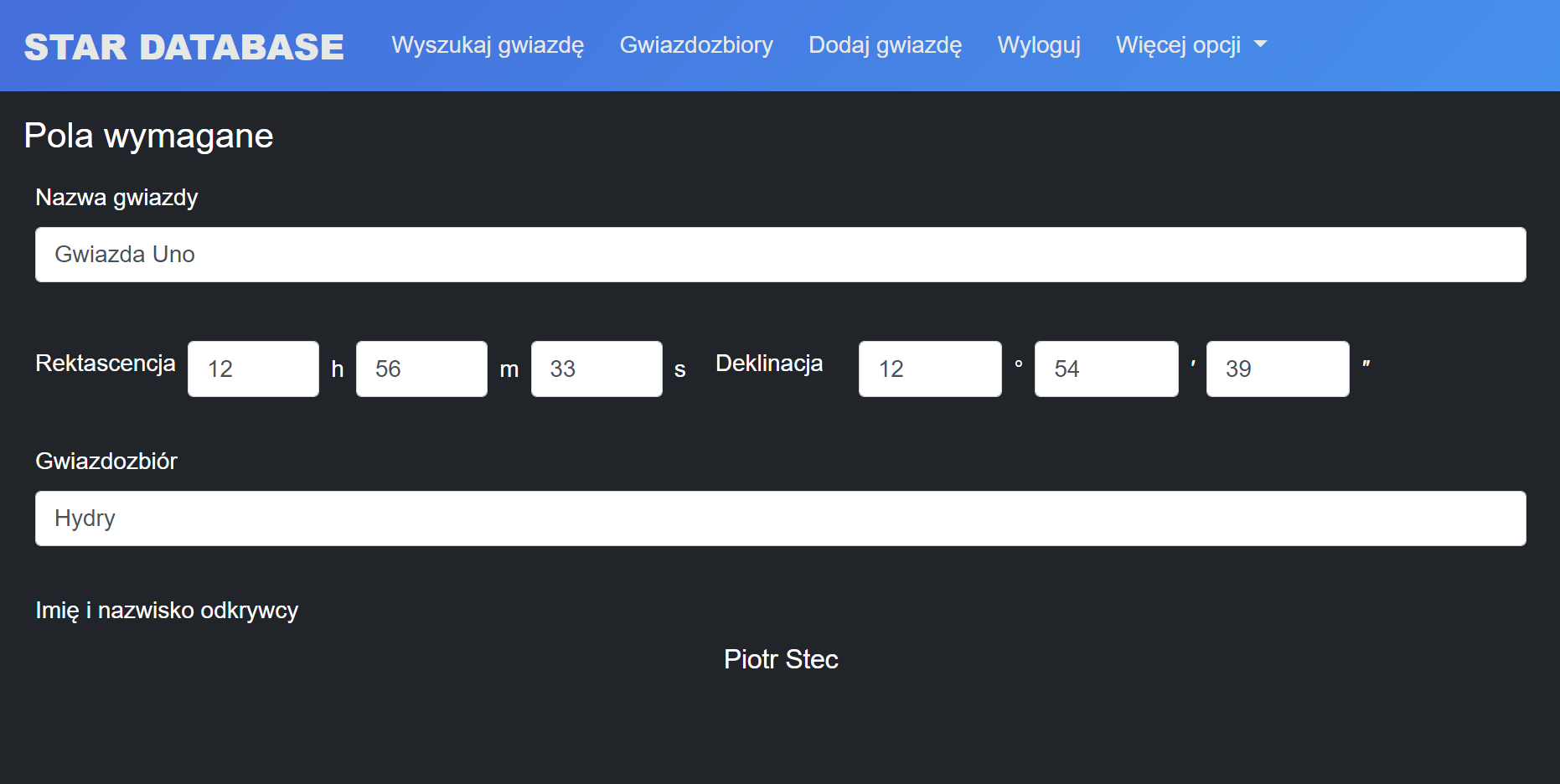
* 1. **Wyszukiwanie gwiazdy**Ta funkcjonalność jest dostępna dla każdej osoby odwiedzającej witrynę. Aby wyszukać gwiazdę należy wpisać jej nazwę w polu wyszukiwania. Po potwierdzeniu wpisanej nazwy użytkownikowi zostaną wypisane nazwy gwiazd najbardziej przypominających wpisaną treść. Drugim sposobem jest wybranie gwiazdy z listy znajdującej się w zakładce dostępnej w informacjach dowolnego gwiazdozbioru.

Rysunek 13: Widok panelu wyszukiwania gwiazdy

* 1. **Gwiazdozbiory**Jednym z elementów wspierających zdobywanie astronomicznej wiedzy jest lista wszystkich gwiazdozbiorów ustanowiona przez Międzynarodową Unię Astronomiczną (eng. International Astronomical Union). Użytkownik może wybrać szukany gwiazdozbiór z wyświetlonej listy.  
     Po wybraniu gwiazdozbioru użytkownik może przejść do strony z szczegółowymi informacjami na temat gwiazdozbioru. Wyświetlane są tam informacje o gwiazdozbiorze takie jak pozycja na niebie liczba gwiazd przypisanych do danego gwiazdozbioru, symbolizm, obraz danego gwiazdozbioru. Szczególną uwagę należy zwrócić na liczbę przypisanych gwiazd do gwiazdozbioru, nie odpowiada ona rzeczywistej liczbie lecz reprezentuję sumę zapisanych w bazie danych obiektów. Liczba ta wraz z czasem działania aplikacji i zaangażowaniem użytkowników będzie systematycznie zwiększana. W tym panelu istnieje możliwość wyświetlenia całej listę gwiazd należących do danego gwiazdozbioru.

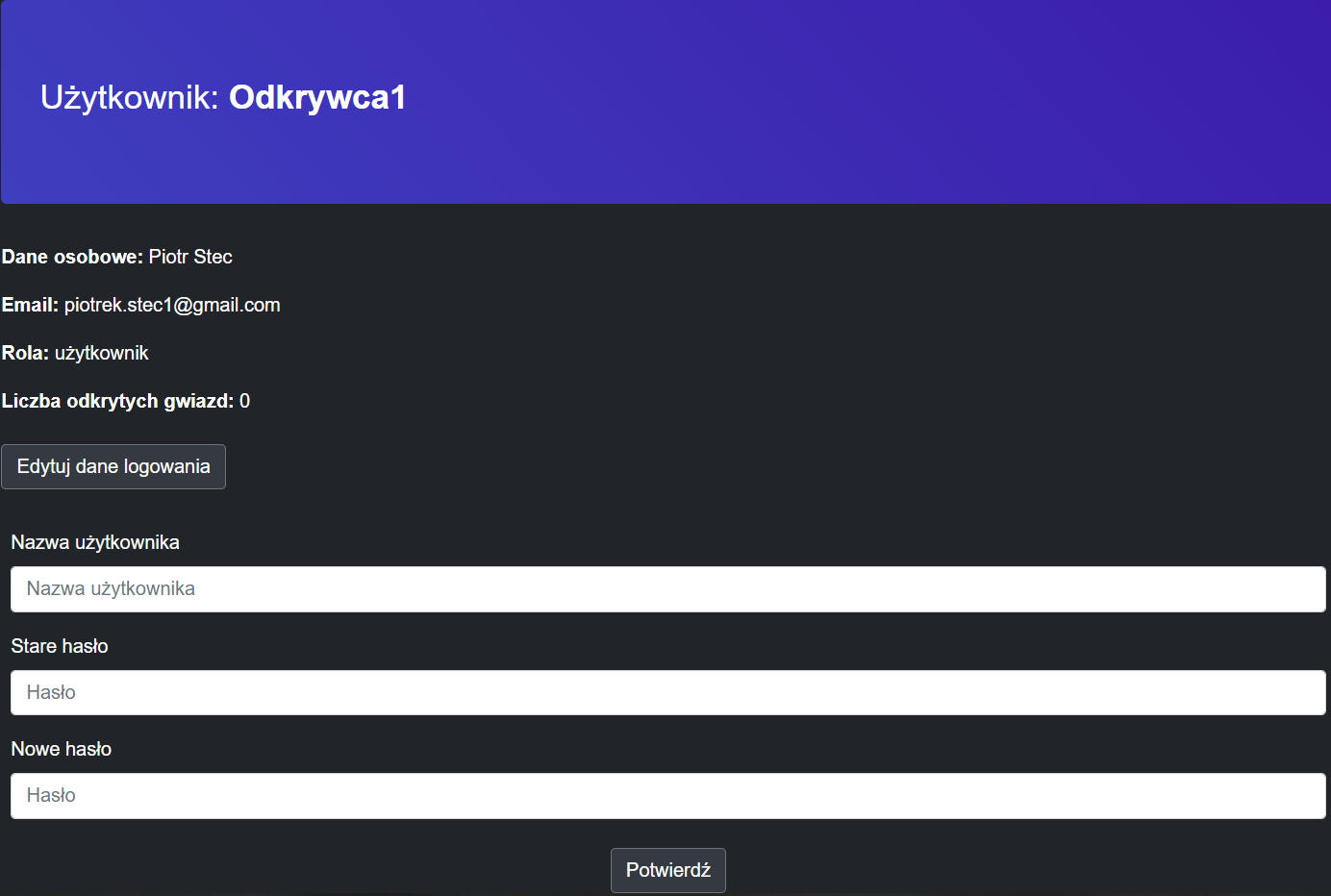
Rysunek 14: Widok panelu gwiazdozbiorów

* 1. **Dodawanie nowej gwiazdy**Opcja ta jest dostępna dla zalogowanego konta. Użytkownik chcąc dodać gwiazdę musi wprowadzić wymagane dane. W aplikacji są one opisane jako „Parametry wymagane”. Własności te są stworzone w celu dokładnej charakterystyki danego obiektu. Posiada je każda gwiazda i każdy jest w stanie je znaleźć bądź zaobserwować. Trudniejsza sytuacja jest w przypadku pozostałych cech gwiazdy, nie każde urządzenie obserwacyjne pozwala na dokładne wyliczenie parametrów takich jak masa bądź typ widmowy gwiazdy, dlatego te wartości są oznaczone jako opcjonalne.  
       
     Każde wprowadzone pole w tej części strony jest sprawdzane pod kontem poprawności fizycznej przez aplikację frontendową jak i backendową.   
     Aplikacja backendowa podczas wywołania endpointu do dodawania gwiazdy, sprawdza czy użytkownik istnieje w bazie danych. Jeśli takowe konto nie istnieje bądź funkcja jest wywołana z źródła nieposiadającego ten sam format danych zostanie zwrócona wiadomość o błędnym formularzu. Nazwa gwiazdy jako podstawowy parametr ją określający musi być unikatowa, jest to sprawdzane w backendzie jak i w samej bazie danych. W przypadku gdy podana nazwa istnieje zwracany jest wyjątek z informacją o już istniejącej nazwie gwiazdy. Wartości charakteryzujące położenie danej gwiazdy na niebie są dwukrotnie sprawdzane przez aplikację, podczas tworzenia formularza w frontendzie oraz po odbiorze danych na backendzie.  
     Rektascencja jako wartość fizyczna musi zawierać się w przedziale od 00h 00m 00s do 24h 00m 00s. Wartości te są sprawdzane poprzez kilka warunków zaaplikowanych w aplikacji frontendowej jak i backendowej.   
     Deklinacja jako reprezentacja wartości fizycznej musi zawierać się w przedziale od -90° 00’ 00’’ do 90° 00’ 00’’. W bazie danych wartości te są podzielone na trzy części z czego dwie pierwsze są zapisywane jako tinyint a trzecia z względu na to, że może posiadać część ułamkowa jako float.  
     Kolejnym ważnym elementem charakteryzującym gwiazdę jest przynależność do gwiazdozbioru. Wpisywana nazwa jest sprawdzana backendowo czy istnieje, w przypadku gdy funkcja nie znajdzie odpowiednika podanej nazwy zostanie zwrócony odpowiedni komunikat informujący o błędzie.   
     Po wszystkich spełnionych krokach zostanie wyświetlony finalny komunikat o wysłaniu formularza oczekującego na potwierdzenie.

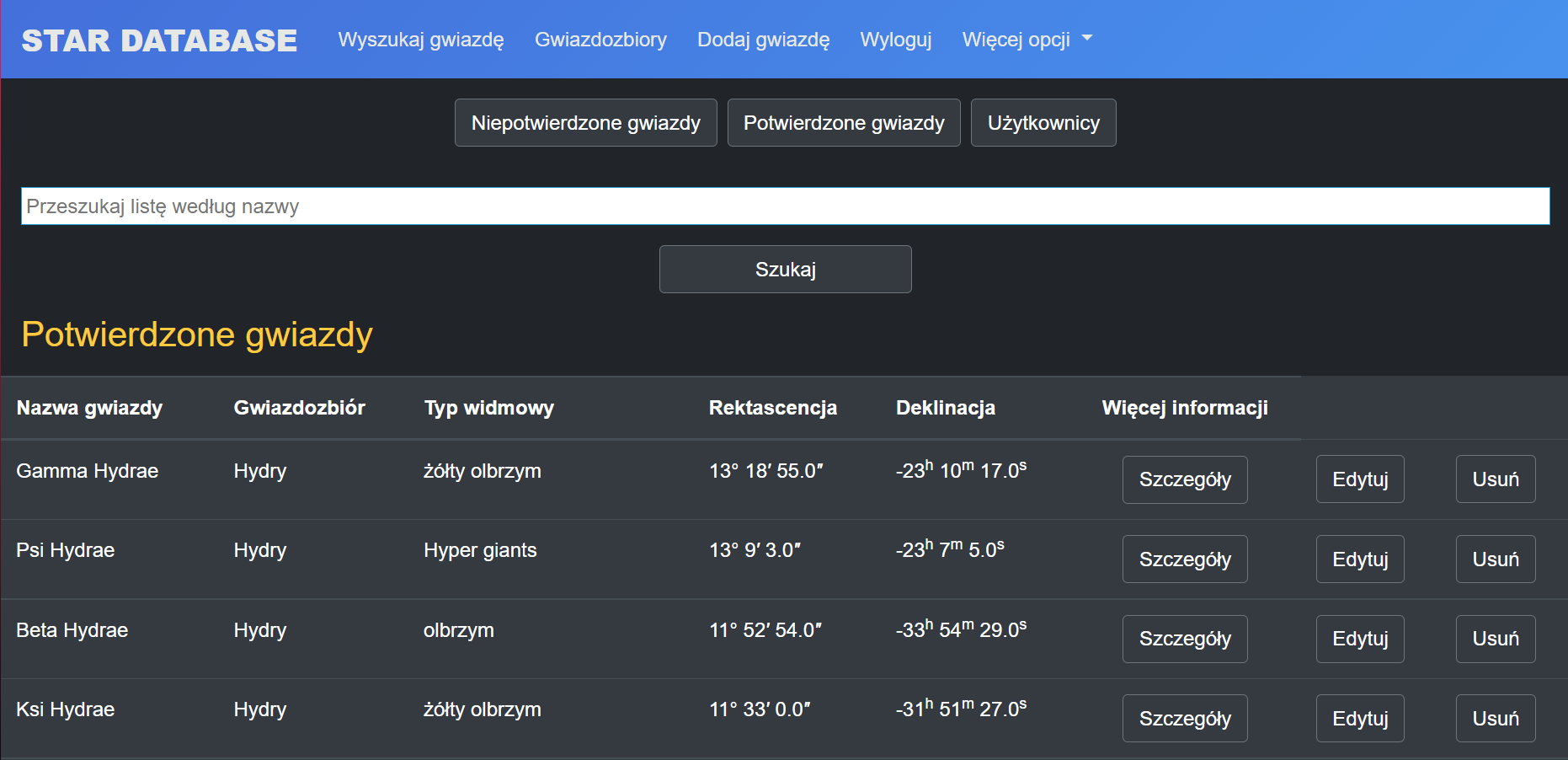
Rysunek 15: Widok panelu dodawania gwiazdy z wymaganymi polami

Rysunek 16: Widok panelu dodawania gwiazdy z opcjonalnymi polami

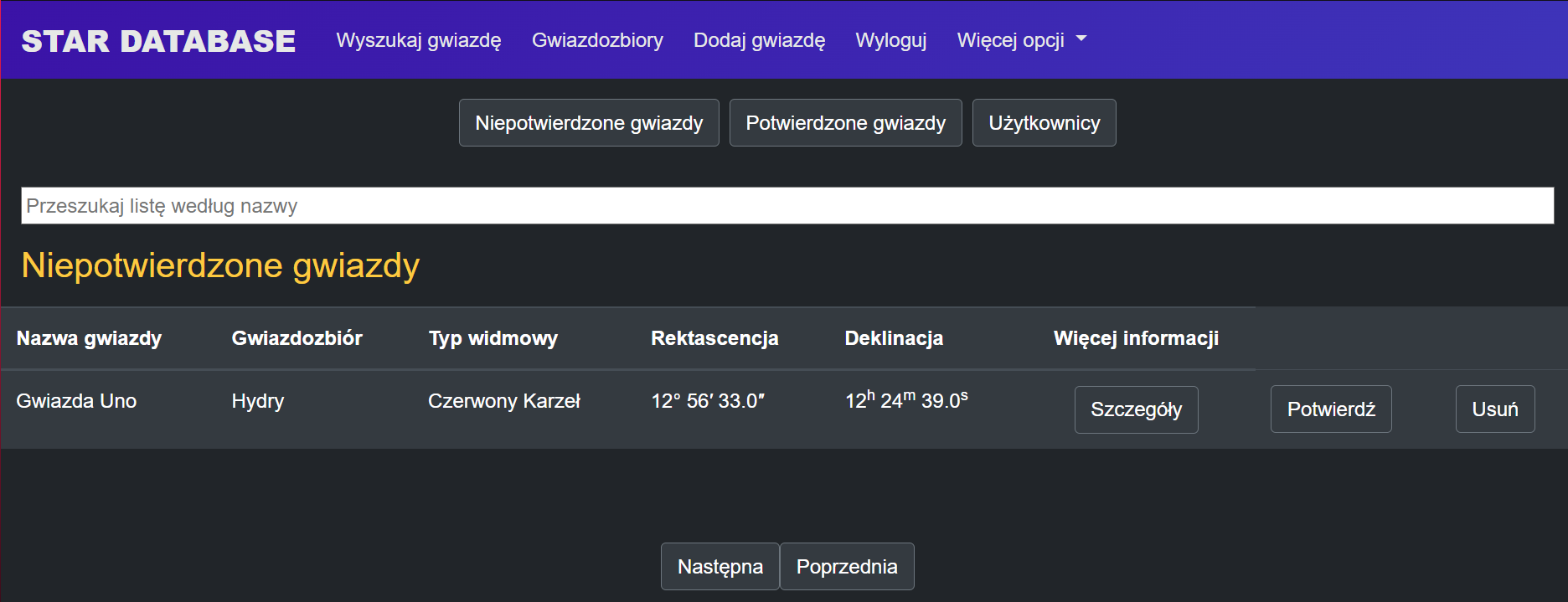
* 1. **Edycja danych logowania**Użytkownik po zalogowaniu ma możliwość zmiany hasła i nazwy użytkownika. Może on tego dokonać po wyświetleniu swojego profilu. Opcja wyświetlenia profilu jest dostępna w pasku zadań. W tym komponencie pokazują się informacje na temat aktualnie zalogowanego użytkownika to znaczy: email, imię, nazwisko, rola spełniana w aplikacji oraz liczba potwierdzonych odkrytych gwiazd. Pod tymi informacjami widnieje przycisk „Edytuj dane użytkownika”, po kliknięciu na ten element wyświetlają się pola, w które użytkownik może wprowadzić nowe dane logowania. Wartości te podlegają kilku warunkom, nowo wpisana nazwa użytkownika nie może istnieć już w bazie danych oraz hasło nie może być krótsze niż osiem znaków. Dodatkowym wymogiem jest wpisanie ponownie nowego hasła. Po poprawnym wprowadzeniu danych zostanie wywołana funkcja w Flask, która sprawdzi poprawność otrzymanych informacji o zalogowanym użytkowniku jak nowych danych logowania. Jeśli wszystkie warunki zostaną spełnione backend wyśle odpowiedź o zmianie danych logowania, która to zostanie wypisana jako komunikat dla użytkownika.

Rysunek 17: Widok profilu użytkownika

* 1. **Edycja danych o gwieździe**Jednym z głównych zadań administratora jest weryfikacja formularzy nowo dodanych gwiazd. Opcja ta jest dostępna jedynie dla upoważnionego konta po zalogowaniu do aplikacji. Przechodząc do listy formularzy administrator może wyświetlić listę nowo dodanych gwiazd oczekujących na potwierdzenie. W tabeli wypisane są podstawowe dane z formularza, administrator może wyświetlić więcej szczegółów przechodząc do zakładki szczegóły. Na końcu każdego wiersza w tej tabeli widnieje przycisk „Potwierdź”. Po jego kliknięciu następuje wywołanie funkcji w Flask, która to sprawdza uprawnienia osoby wysyłającej dane. Kolejnym krokiem jest edycja rekordu w bazie danych, poprzez zmianę wartości z „NO” na „YES”. Po tym zapytaniu zostanie zwrócona informacja do Flaskm, a następnie wyświetlona w części frontendowej.

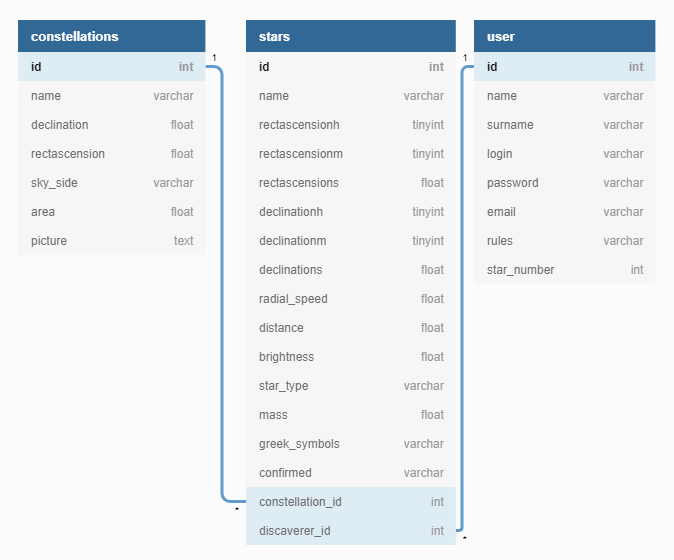
Rysunek 18: Widok listy potwierdzonych gwiazd w aplikacji

* 1. **Potwierdzenie gwiazdy**Jednym z głównych zadań administratora jest weryfikacja formularzy nowo dodanych gwiazd. Opcja ta jest dostępna jedynie dla upoważnionego konta po zalogowaniu do aplikacji. Przechodząc do listy formularzy administrator może wyświetlić listę nowo dodanych gwiazd oczekujących na potwierdzenie. W tabeli wypisane są podstawowe dane z formularza, administrator może wyświetlić więcej szczegółów przechodząc do zakładki szczegóły. Na końcu każdego wiersza w tej tabeli widnieje przycisk „Potwierdź”. Po jego kliknięciu następuje wywołanie funkcji w Flask, która to sprawdza uprawnienia osoby wysyłającej dane. Kolejnym krokiem jest edycja rekordu w bazie danych, poprzez zmianę wartości z „NO” na „YES”. Po tym zapytaniu zostanie zwrócona informacja do Flaskm, a następnie wyświetlona w części frontendowej.

Rysunek 19: Widok listy niepotwierdzonych gwiazd

* 1. **Usuwanie gwiazdy**Jedna z opcji wchodząca w skład edycji rekordów w bazie danych, dostępna dla administratora to usuwanie obiektu. Jest ona dostępna jedynie dla zalogowanego administratora, przechodząc za pomocą paska zadań do listy formularzy użytkownik dostaje dwie listy, potwierdzonych oraz niepotwierdzonych gwiazd. Dla tych dwóch tabel jest dostępna opcja usuwania. Ta opcja nie jest dostępna dla tabeli z listą zarejestrowanych użytkowników. Administrator w celu usunięcia obiektu musi kliknąć przycisk „Usuń” czym wywołuje funkcję w Flask, zostają tam sprawdzone uprawnienia osoby wysyłającej dane. Po poprawnej weryfikacji następuje wysłanie zapytania do bazy danych o usunięcie wskazanego rekordu. W przypadku gdy gwiazda posiadała odkrywcę dla tego konta zostanie zmniejszona liczba odkrytych gwiazd. Po poprawnym wykonaniu operacji zostanie zwrócona informacja na backend aplikacji, tam przekazana do frontendu.
  2. **Nadawanie/Odbieranie roli administratora**

1. **Baza danych oraz jej obsługa**
   1. **MySQL oraz MariaDB**Baza danych składa się z trzech tabeli. W tabeli „user” przechowywane są informacje o osobach posiadających konto w aplikacji są to imię, nazwisko, nazwa użytkownika, hasło, mail, rola w apliakcji oraz liczba odkrytych gwiazd. Kolumna „password” przechowuje wartość hash utworzoną za pomocą metody w pythonie. Dwie kolumny „login” oraz „email” są określone jako wartości unikatowe oznacza to, że w bazie danych nie mogą istnieć dwie identyczne pozycje w tym samym czasie.   
      W tabeli „star” przechowywane są wartości charakteryzujące gwiazdę takie jak nazwa, rektascencja oraz deklinacja, które są podzielone na trzy fragmenty, prędkość radialna, dystans, jasność, typ widmowy, masa, symbol grecki, id gwiazdozbioru, id odkrywcy, status weryfikacji. Nazwa gwiazdy jest określona jako wartość unikatowa pozwala to uniknąć błędów związanych z wyszukiwaniem danych oraz ogólnie przyjętą praktyką w astronomii. Wartości rektascencja oraz deklinacja są podzielone na trzy elementy w celu uproszczenia logistycznego podejścia do liczb jakie mogą przyjmować. Kolumna id gwiazdozbioru odnosi się do istniejącej pozycji w bazie danych w tabeli „constellations”, wymogiem istnienia gwiazdy w bazie danych jest jej przynależność do gwiazdozbioru. Wartość id odkrywcy jest opcjonalna jest to spowodowane danymi, które zostały tam zapisany podczas tworzenia aplikacji. Składają się na nie gwiazdy tworzące szkic danego gwiazdozbioru. Kolumna „confirmed” służy do rozróżnienia gwiazd, które zostały wysłane przez użytkowników i oczekują na potwierdzenie oraz tych już potwierdzonych. Liczba rekordów w tej tabeli będzie się zwiększać wraz z działaniem aplikacji.   
      W tabeli „constellations” widnieją kolumny przechowujące następujące wartości: nazwa, deklinacja, rektascencja, symbolizm, półkola nieba, powierzchnia oraz obraz. Żaden użytkownik korzystający z aplikacji nie ma dostępu do edycji bądź dodawania danych w tej tabeli. Liczba rekordów w tej tabeli reprezentuje rzeczywisty podział nieba przez Międzynarodową UnięAstronomiczną, jest to dokładnie 88 pozycji. Każdy gwiazdozbiór posiada krótki opis jego znaczenia w kulturze oraz odnośnik do obrazka zapisanego w frontendowej części aplikacji.   
      Baza danych jest udostępniana poprzez serwis MariaDB udostępniający bazę danych w obrębie wybranej sieci. Jest on bardzo wygodny i prosty w użyciu, aby z niego korzystać wystarczy podczas instalacji wybrać opcję jako usługi spowoduje to automatyczny rozruch po włączeniu maszyny na której zamierzamy pracować. Istnieje druga opcja, która to po zainstalowaniu wymaga wpisania jednej komendy w terminalu oraz wpisania hasła. Do konfiguracji jak i tworzenie bazy danych został wykorzystany program HeidiSQL. Jest on polecany do pracy z MariaDB, pozwala on na tworzenie, edycję i zarządzanie bazą danych.

Rysunek 20: Schemat tabel w bazie danych

* 1. **SQLAlchemy oraz MariaDB**W części backendowej aplikacji do wykonywania operacji na bazie danych wykorzystana jest biblioteka SQLALchemy. Tworzy ona połączenie z bazą danych oraz pozwala na tworzenie metod działających na mapowaniu obiektowo-relacyjnym. W strukturze projektu znajduje się katalog „orm” zwiera on plik python, którego zadaniem jest nawiązanie połączenia z bazą danych. Plikiem, który najczęściej odwołuje się niego jest „orm.py”, służy on do odzwierciedlenia istniejącej bazy danych poprzez jej mapowanie na obiekty. Każda tabela w bazie danych musi posiadać swój odpowiednik. Plik podzielony jest na klasy, gdzie jedna klasa reprezentuje jedną tabelę w tym miejscu także określane są wartości jakie przechowują kolumny. Następnym krokiem do korzystania z biblioteki jest utworzenie plików posiadających metody do zarządzania bazą danych. W aplikacji istnieją cztery pliki, „db.py” jest to nadrzędny plik określający możliwe do wykorzystania w programie metody. Pozostałe trzy przechowują metody stworzone do edycji konkretnej tabeli.   
     Poprzez odwołanie do konkretnej funkcji w pliku jesteśmy w stanie edytować dane w bazie. Część metod w tych plikach posiada dodatkowe zabezpieczenia chroniące przed próbą zapisu nieoprawnych danych.

1. **Podsumowanie**Aplikacja jest przeznaczona dla osób zaczynających zdobywać wiedzę na temat kosmosu jak i pozwoli im czynnie uczestniczyć w jego odkrywaniu. Zadanie to nie jest łatwe, ale dzięki prostemu przedstawieniu rzeczy często trudnych i skomplikowanych może zachęcać użytkowników do głębszego zrozumienia otaczającego nas świata.  
   Aplikacja z punktu działania informatyki działa na trzech warstwach, są one tak ze sobą połączone, aby można było je nazwać poprawnie działającym serwisem. Przewodnią myślą podczas tworzenia bazy danych było połączenie więcej niż jednego obiektu astronomicznego.   
   Po stronie użytkownika przedstawia ona najważniejsze elementy pomagające w obserwacji nocnego nieba. Przewidziane konto z zwiększoną liczbą uprawnień pozwoli na swobodny rozwój aplikacji. Witryna pokazuje nowatorski sposób na działanie serwisu, w którym ważnymi elementami jest uczestnictwo osób oraz poprawność wartości zapisywanych w bazie danych.
2. **Bibliografia**
   1. „Programming Python: Powerful Object-Oriented Programming” Mark Lutz
   2. „Ng-Book 2: the Complete Book on Angular 2 (Volume 2)” Nate Murray, Ari Lerner, Felipe Coury, Carlos Taborda
   3. „Learning MySQL and MariaDB: Heading in the Right Direction with MySQL and MariaDB” Russel J. T. Dyer
3. **Wykaz rysunków**
4. **Streszczenie**Aplikacja została stworzona do pomocy przy wyszukiwaniu informacji na temat gwiazd i gwiazdozbiorów. Składa się ona z trzech części frontendu napisanego przy pomocy frameworka Angulkar, backendu stanowiącego połączenie bibliotek Flask oraz SQLAlchemy stworzonych dla języka Python oraz bazy danych udostępnianej przy pomocy serwisu MariaDB w języku MySQL. Głównym zadaniem aplikacji jest pokazanie podstawowych informacji o gwiazdach i gwiazdozbiorach. Dodatkowym aspektem jest możliwość wyszukania oraz wyświetlania szczegółowych informacji na temat tych obiektów. Szczególnym wyróżnikiem jest to, że internauci mogą przyczynić się do rozwoju aplikacji. Każdy zarejestrowany użytkownik posiada możliwość wysłania formularza z informacjami o gwieździe, którą odkrył bądź, która nie widnieje w bazie danych. W celu kontroli przepływu informacji w aplikacji przewidziana jest rola administratora. Posiada on podstawowe możliwości edycji i usuwania. Głównym jego zadaniem jest weryfikowanie nowo przesłanych formularzy. Jako, że napływ formularzy może być zbyt duży dla jednego administratora może on dla osób wyróżniających się aktywnością bądź potwierdzonych przez inne drogi komunikacji, nadać prawa administratora. Aplikacja za cel obrała pokazanie, że wiedza o gwiazdach nie wymaga żadnych wyższych tytułów naukowych bądź wieloletniej eksploracji informacji o kosmosie.