Baza Danych

Sklep z produktami telekomunikacyjnymi oraz oprogramowaniem

TelcoWrocław

Prowadzący:

Mgr. Inż. Bartosz Majorowski

Projektant:

Paweł Koryciński

Grupa A

6749

Spis treści:

1. Opis biznesowy „świata rzeczywistego”
2. Lista wymagań funkcjonalnych
3. Lista wymagań niefunkcjonalnych
4. Lista przypadków użycia
   1. Sprint 1
   2. Sprint 2
   3. Sprint 3
5. Diagram sekwencji
6. Diagram klas
   1. Warstwa logiki biznesowej (Kontroler)
   2. Warstwa niskopoziomowa
   3. Warstwa danych ( Baza danych )
   4. Warstwa prezentacji ( Interfejs użytkownika )
   5. Pełna architektura
7. Weryfikacja ryzyka
8. Dodatkowe linki

I. Opis biznesowy „świata rzeczywistego” w języku klienta – aplikacja „Utrzymanie i zarządzanie urządzenia typu RET”

1. Opis zasobów ludzkich

* Pracownik
  + przygotowuje środowisko uruchomieniowe
    - instalacja
      * systemu operacyjnego
      * potrzebnych zależności w postaci oprogramowania oraz sterowników
  + weryfikuje
    - poprawność podłączenia urządzenia do komputera oraz źródła zasilania
    - zgodność urządzenia ze standardem AISG2.0
  + administruje i zarządza aplikacją „O&M RetDriver”
    - identyfikuje port do którego podłączone jest urządzenie
    - inicjuje proces adresacji urządzenia
      * reaguje na ewentualne błędy zwracane podczas komunikacji z urządzeniem
    - wywołuje wysokopoziomowe komendy w celu interakcji ze sprzętem
      * subskrypcja na alarmy
      * po ukończonym procesie ustawienia kąta, ustanawia proces cyklicznego podtrzymywania rozmowy z urządzeniem
* Klient
  + dostarcza
    - zakupione wcześniej urządzenie typu RET od dowolnego producenta
      * zalecane jest urządzenie firmy Katherine
      * obowiązkiem jest dostarczenie specyfikacji urządzenia otrzymywanej wraz z zakupem
    - docelowy kąt anteny zdefiniowany w stopniach
  + opcjonalnie zakupuje licencję na wsparcie dla zainstalowanej konfiguracji
    - weryfikacja logów z działania urządzenia
    - walidacji poprawności stanu urządzenia zaraportowanego w bazie danych
    - możliwość pojawienia się w miejscu urządzenia w przeciągu ustalonego wcześniej czasu

1. Przepisy

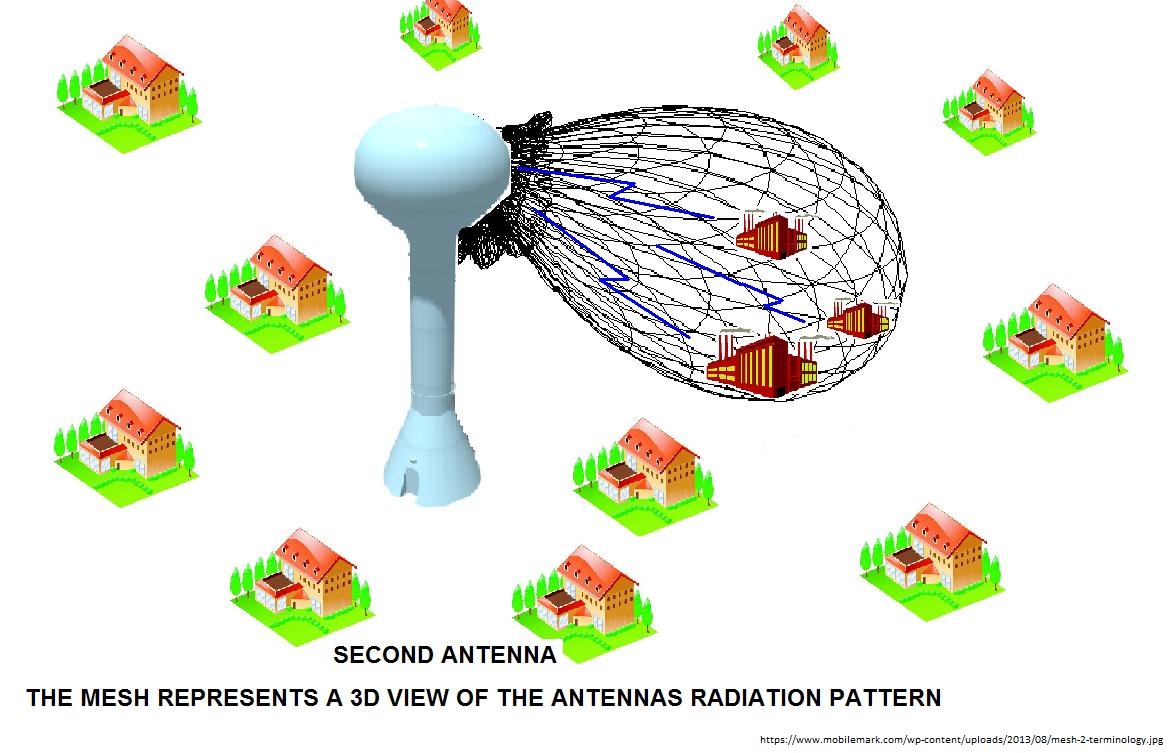
System ma zadanie umożliwienie komunikacji z urządzeniem typu RET (Remote Electrical Tilt) w celu wykonywania na nim operacji zdefiniowanych przez protokół AISG2.0. W zależności od zapotrzebowania użytkownika powinien on umożliwiać: dostarczenie napięcia lub/i korzystanie z jego dobrodziejstw: ustawienie kąta nachylenia anteny dla technologii LTE/5G.

Weryfikacja aktualnego stanu urządzenia na błędy krytyczne bądź pomniejsze będzie odbywała się co 2 minuty. W przyszłości planowany jest system automatycznej próby reakcji na znane typy błędów możliwe do rozwiązania bez konieczności obecności w miejscu jego instalacji.

1. Dane techniczne

Urządzenie typu RET podłączane jest do komputera poprzez specjalny kabel RS485-USB2.0, otrzymywany wraz z zakupem oprogramowania. Sprzęt wymaga zasilania 30V, które dostarczymy poprzez dodatkowy rozgałęźnik w wyżej wspomnianym kablu.

1. Lista wymagań funkcjonalnych (wraz z minimalnym zestawem atrybutów)
   1. Odczytanie aktualnego stanu dowolnego urządzenia typu RET
      * Po odłączeniu oraz podłączeniu urządzenia, nie chcielibyśmy być zmuszeni do tego aby zapisywać na kartce szereg koordynatów bądź parametrów, w związku z czym jednym z pierwszym kroków jest pobranie faktycznego stanu urządzenia
      * Po wysłaniu wiadomości której zawartość definiuje chęć bardziej szczegółowych odpowiedzi na każdą komendę ( zwrócenie zaistniałych błędów w trakcie działania urządzenia ), możemy na bieżąco być informowani o ewentualnych problemach sprzętowych/oprogramowania zakupionego RET-a
   2. Adresacja urządzenia
      * Urządzenie po podłączeniu do prądu oraz do portu szeregowego znajduje się w stanie NoAddress to znaczy posiada adres równy 0. Aby rozpocząć wykonywanie wysokopoziomowych komend, należy dopasować się do maski urządzenia tworzonej z „SerialNumber” oraz „VendorCode” urządzenia. Zakładając, że algorytm przeszukiwania binarnego poprawnie ukończył procedurę odnajdywania maski, urządzenie otrzyma adres 0x02.
   3. Kalibracja urządzenia
      * Po udanym zestawieniu warstwy drugiej, RetDriver może rozpocząć interakcję z urządzeniem na „wyższym poziomie” tzn. zacząć wykonywać komendy na poziomie abstrakcji silnika przy pomocy którego można ustawić kąt. Zalecaną procedurą przed ustawieniem kąta jest przeprowadzenie procedury kalibracji. Polega ona na 3 minutowej procedurze w trakcie której silnik wykona pracę na całym swoim zakresie, tzn. ustawianie kąta w zakresie (0;180) a następnie (180; 0). Jeśli po wykonaniu tej procedury urządzenie odpowie wartością 0x00 to znaczy, że urządzenie jest sprawne.
   4. Ustawienie dowolnego kąta wspieranego przed producenta
      * W zależności od trybu ustawienia anteny, RET może zmienić kierunek oraz szerokość wiązki fizycznie bądź elektrycznie. Konieczność zmiany kąta wynika z faktu iż ludzie przemieszczają się z miejsca na miejsce a naszym celem jest pokrycie jak najlepszym zasięgiem obszaru gdzie znajduje się jak najwięcej osób. Procedura jest na tyle bezpieczna iż jeśli zapomnieliśmy wcześniej skalibrować urządzenia, to podczas próby jej wykonania, odpowie nam, że jest nieskalibrowane.



1. Lista wymagań niefunkcjonalnych (do opracowania)
   1. Ograniczenie wydajnościowe
      * Czas po którym urządzenie posiada ustawiony zadany przez użytkownika kąt to 6 minut
      * Jedna instancja produktu O&M RetDriver może obsługiwać maksymalnie jedno urządzenie
      * Jedno urządzenie może być obsługiwane jedynie przez jedną instancję aplikacji O&M RetDriver
      * W momencie braku ustanowienia cyklicznej rozmowy z urządzeniem, po 3 minutach traci swój adres oraz zostaje przerwana komunikacja z nim
   2. Czy jest wymagany masowy dostęp (Internet)?
      * Nie aczkolwiek planowany jest klient internetowy do monitoringu i zarządzania
   3. Proponowane technologie
      * OS - Manjaro Linux 17.01
2. Lista przypadków użycia - propozycja

Sprint 1

|  |
| --- |
| Przypadki użycia |
| 1. PU Adresacja urządzenia   Jako operator chcę zaadresować urządzenie wartością 0x02, abym mógł rozpocząć wykonywanie wysokopoziomowych procedur |
| 1. PU Reset urządzenia   Jako operator chcę zresetować urządzenie, abym mógł łatwo reprodukować błędne scenariusze oraz zweryfikować czy zadane przeze mnie parametry zostały zapisane. |

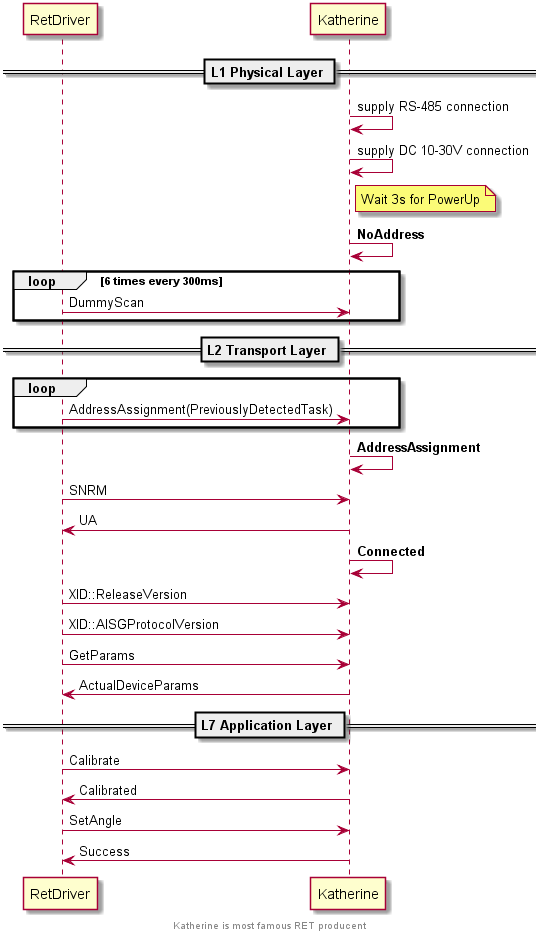
Sprint 2

|  |
| --- |
| Przypadki użycia |
| 1. PU Kalibracja   Jako operator chcę przeprowadzić kalibrację urządzenia, abym mógł przejść do 4 PU. |
| 1. PU Ustawienie kąta   Jako operator chce ustawić dowolny poprawny kąt, abym mógł zmienić obszar pokryty zasięgiem sieci telekomunikacyjnej |

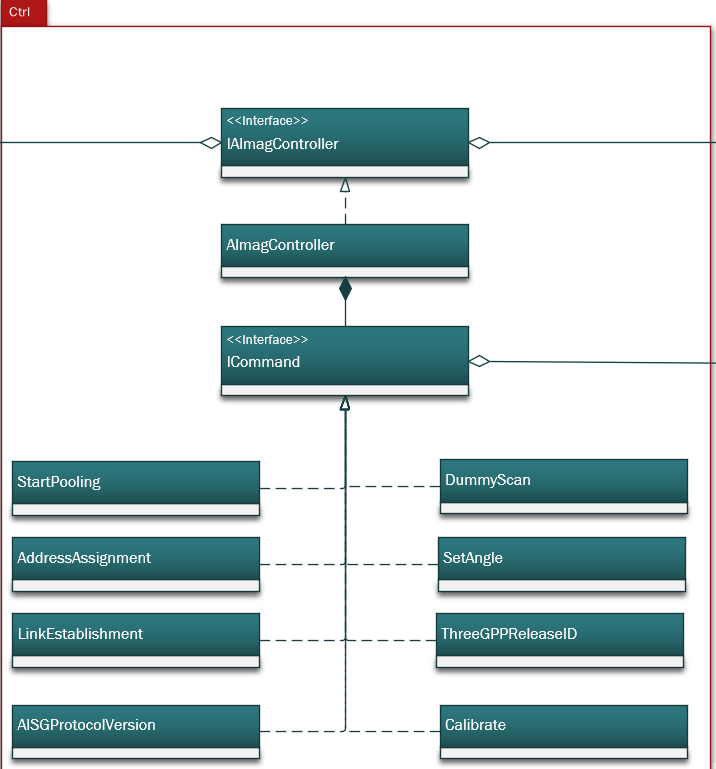
Sprint 3

|  |
| --- |
| Przypadki użycia |
| 5. PU Rozpoznanie alarmów raportowanych na żądanie przez urządzenie  Jako operator chcę otrzymać informację o tym, że silnik nie działa poprawnie, aby mógł niezwłocznie wykonać akcje diagnostyczno-naprawcze. |
| 6. PU Uruchomienie cyklicznego podtrzymywania rozmowy z urządzeniem  Jako operator chce mieć możliwość pozostawienie zaadresowanego urządzenia na czas dłuższy niż 3 minuty, abym mógł zająć się innymi procedurami bądź odejść od komputera. |

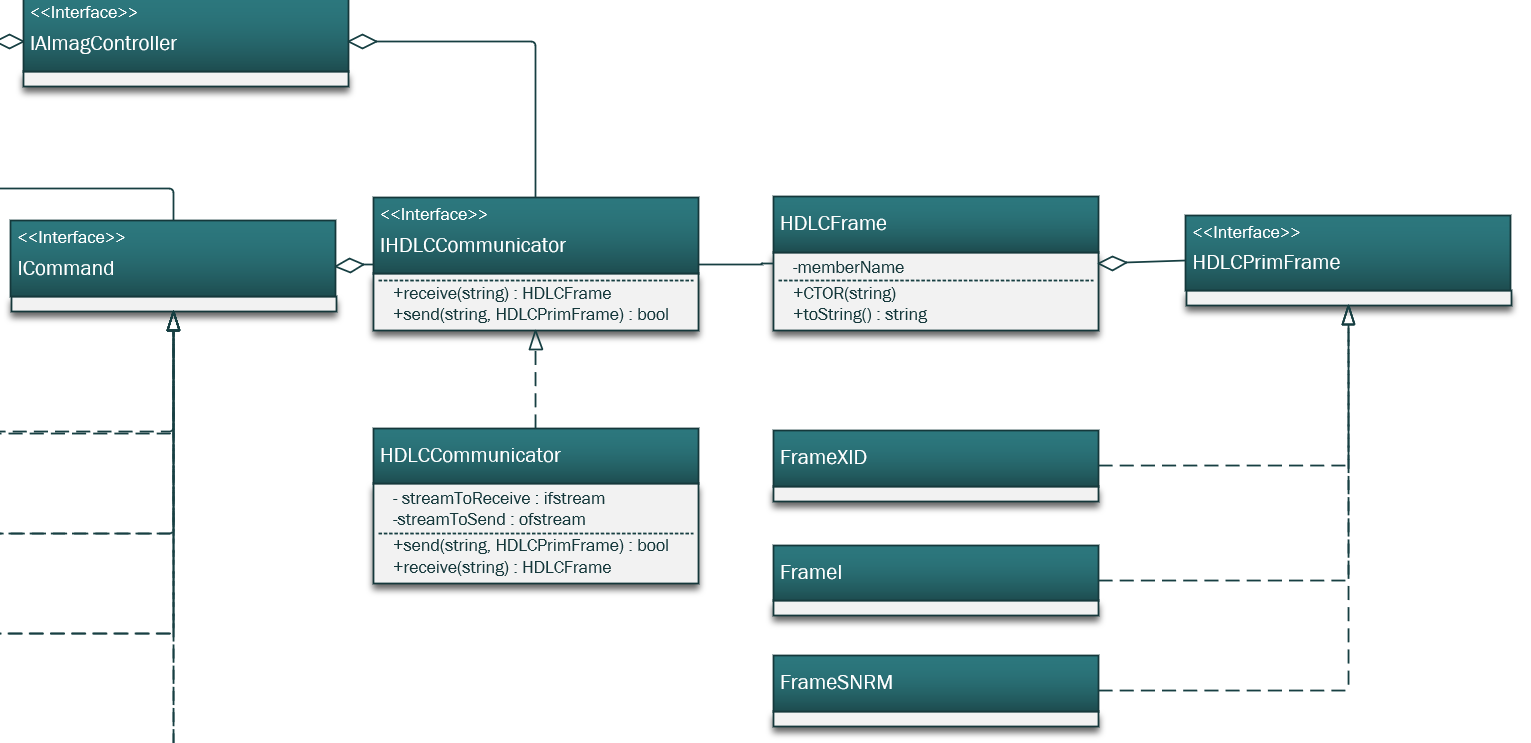
1. Diagram sekwencji



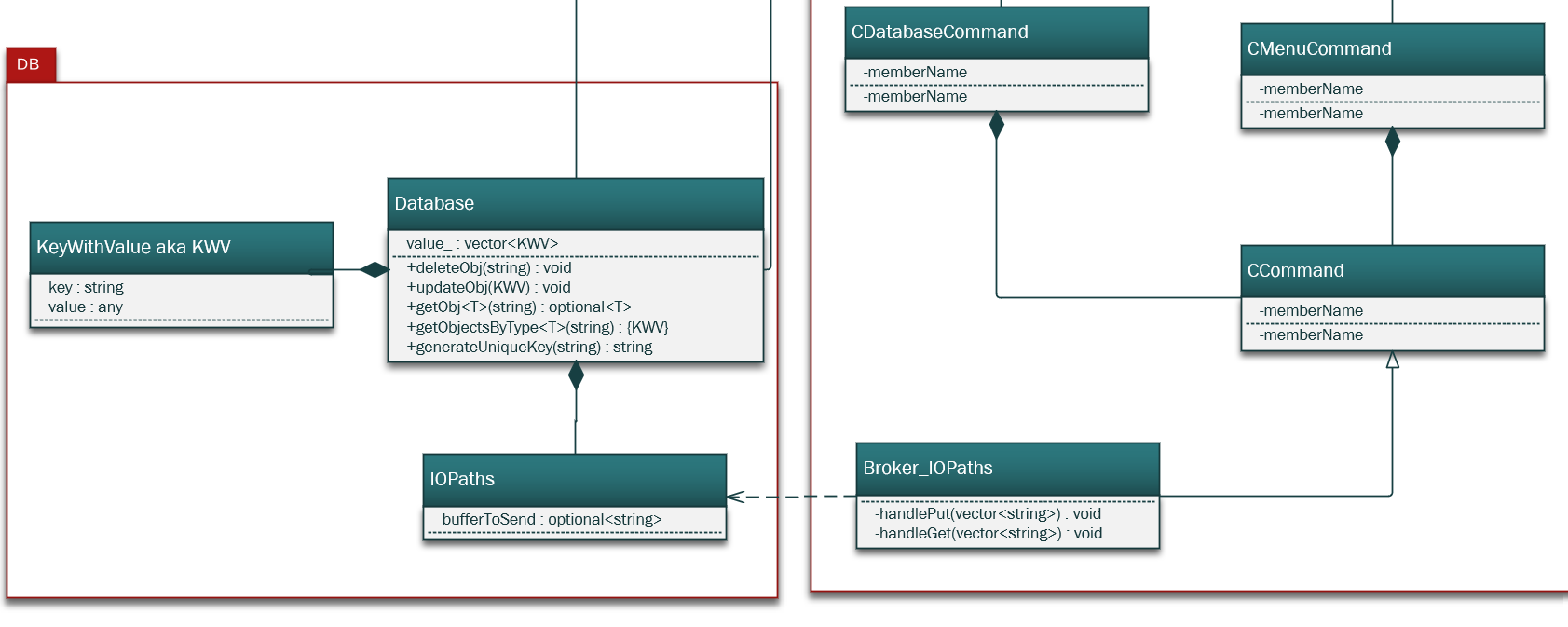
1. Diagram klas
   1. Warstwa logiki biznesowej ( Kontroler )



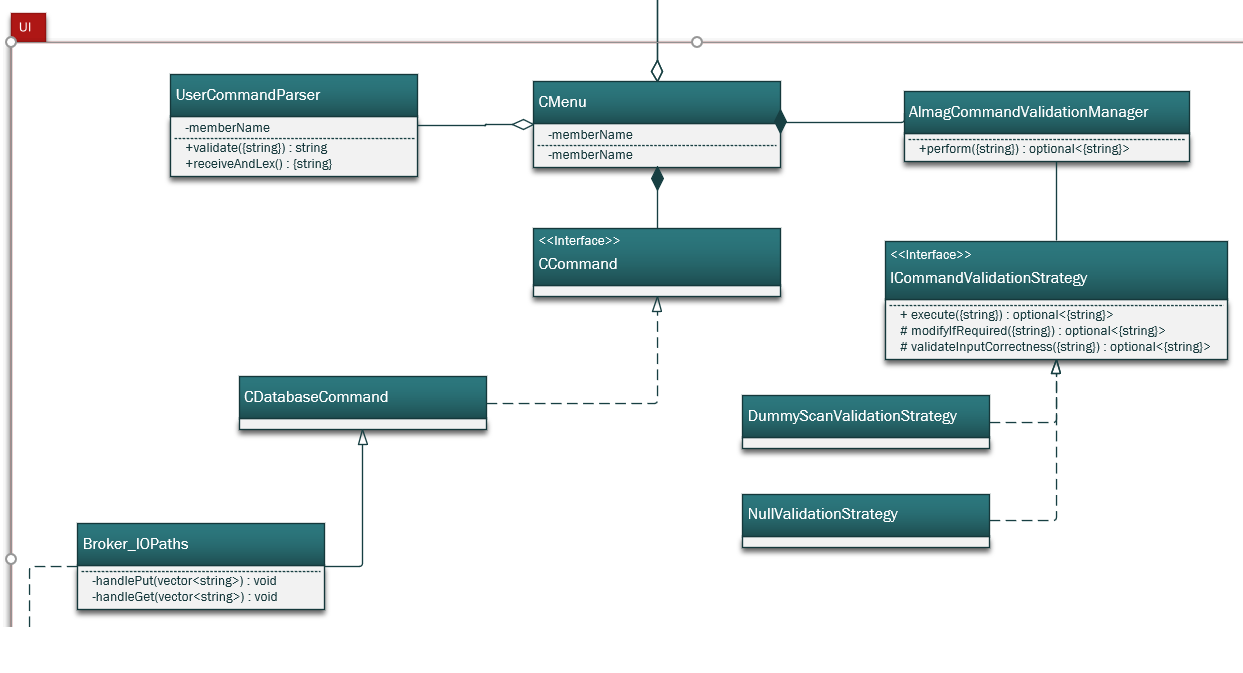
* 1. Warstwa niskopoziomowa



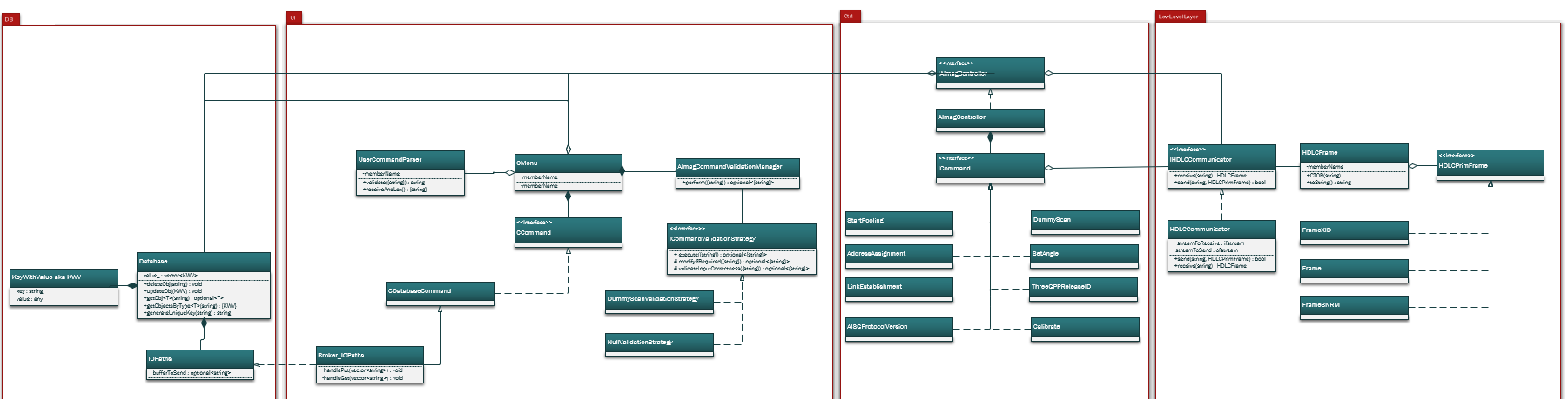
* 1. Warstwa danych ( Baza danych )



* 1. Warstwa prezentacji ( Interfejs użytkownika )



* 1. Pełna architektura



1. Weryfikacja ryzyka
   1. Podczas próby implementacji systemu zgodnie z protokołem AISG2.0 trzeba wziąć pod uwagę iż niestety nie wszystkie urządzenia działają zgodnie ze standardem, a więc w przypadku braków wiedzy na temat oczekiwań wobec niego, można zajść w ślepy zaułek z którego wyjściem jest negatywna wiadomość w sprawie próby pełnej obsługi urządzenia RET
   2. Trzeba trzymać balans pomiędzy zmierzaniem do fizycznego połączenia się z urządzeniem a operowaniu na jego symulatorze, dlaczego? Testy jednostkowe/modułowe będą trwały znacznie dłużej na prawdziwym sprzęcie co wydłuży znacznie wprowadzanie nowych funkcjonalności oraz poprawianie już istniejących.
2. Dodatkowe linki
   1. Diagram UML
      * <https://github.com/CopernicusSemestr5/Almag_RetDriver/tree/UML>
   2. Kod produkcyjny opatrzony licencją (CC BY-NC-ND 4.0)
      * <https://github.com/CopernicusSemestr5/Almag_RetDriver>