

Dokumentacja projektu zaliczeniowego

Przedmiot: Inżynieria oprogramowania

Temat: **Aplikacja do automatyzacji ustawień optycznych urządzeń dla operatorów filmowych**
Autorzy: **Jakub Wojciechowski, Jan Biskupski**
Grupa: I1-210A
Kierunek: informatyka
Rok akademicki: 2
Poziom i semestr: I/4
Tryb studiów: stacjonarne

1 Spis treści

2	Odnośniki do innych źródeł.....	4
3	Słownik pojęć	5
4	Wprowadzenie	6
4.1	Cel dokumentacji.....	6
4.2	Przeznaczenie dokumentacji	6
4.3	Opis organizacji lub analiza rynku.....	6
4.4	Analiza SWOT organizacji	7
5	Specyfikacja wymagań	8
5.1	Charakterystyka ogólna.....	8
5.2	Wymagania funkcjonalne.....	9
5.3	Wymagania niefunkcjonalne.....	40
6	Zarządzanie projektem	42
6.1	Zasoby ludzkie	42
6.2	Harmonogram prac.....	42
6.3	Etapy/kamienie milowe projektu	42
7	Zarządzanie ryzykiem.....	44
7.1	Lista czynników ryzyka	44
7.2	Ocena ryzyka.....	44
7.3	Plan reakcji na ryzyko	44
8	Zarządzanie jakością.....	46
8.1	Scenariusze i przypadki testowe	46
9	Projekt techniczny	72
9.1	Opis architektury systemu.....	72
9.2	Technologie implementacji systemu.....	73
9.3	Diagramy UML	73
9.4	Charakterystyka zastosowanych wzorców projektowych.....	78
9.5	Projekt bazy danych	79
9.6	Projekt interfejsu użytkownika.....	79
9.7	Procedura wdrożenia	95
10	Dokumentacja dla użytkownika.....	96
11	Podsumowanie	97
11.1	Szczegółowe nakłady projektowe członków zespołu	97

12	Inne informacje	Błąd! Nie zdefiniowano zakładki.
----	-----------------------	---

2 Odnośniki do innych źródeł

- Zarządzania projektem – sugerowane JazzHub
- Wersjonowanie kodu – sugerowany Git (hosting np. na Bitbucket lub Github), ew. SVN
- System obsługi defektów – np. Bitbucket, JazzHub

3 Słownik pojęć

- Użytkownik darmowy – użytkownik, który nie wykupił licencji, ani subskrypcji.
- Użytkownik premium (z wykupioną licencją) – użytkownik, który wykupił licencję, ale nie wykupił subskrypcji.
- Użytkownik premium (z wykupioną subskrypcją) - użytkownik, który wykupił subskrypcję. Ma dostęp do wszystkich zalet zakupu licencji, a nawet synchronizację ustawień z chmurą.
- Administrator – osoba, która zarządza serwerem, użytkownikami, aplikacjami etc. Może modyfikować wszystko.

4 Wprowadzenie

4.1 Cel dokumentacji

Celem dokumentacji jest zaprezentowanie przeznaczenia aplikacji oraz zakresu jej działania. Niniejsza dokumentacja zawiera techniczny opis aplikacji do automatyzacji ustawień optycznych urządzeń dla operatorów filmowych, określa sposób jej użytkowania, prezentuje zaimplementowane funkcjonalności i określa grupy użytkowników końcowych.

4.2 Przeznaczenie dokumentacji

Dokumentacja jest przeznaczona dla użytkowników końcowych, czyli osób pracujących przy produkcji filmowej. Omawia ona sposób używania tego oprogramowania, poszczególnych komponentów aplikacji, czyli jakie kamery są wspierane, co jest potrzebne do stworzenia akcesoriów np. follow focus, oraz te, które są wspierane domyślnie przez program. Również prezentuje techniczny opis funkcjonalności działania aplikacji i jej komponentów.

4.3 Opis organizacji lub analiza rynku

Jedna z dwóch opcji:

1. Jeśli **dla konkretnej organizacji**: Czym jest organizacja, dla której realizowany będzie system; jak działa (lub będzie działała), kiedy system będzie wdrożony – tutaj nie odwołujemy się do samego systemu, tylko opisujemy samo działanie organizacji i role. W szczególności – jak wyglądają główne procesy biznesowe.
2. Jeśli na masowy rynek: Pobieźna analiza rynku. Dla kogo będzie przydatny taki system. Ile jest organizacji, które będą mogły z niego skorzystać, użytkowników w poszczególnych organizacjach. Czy te organizacje stanowią jednorodną grupę czy są różne rodzaje. Co one mają ze sobą wspólnego. Jak ta liczba będzie się zmieniała w najbliższej przyszłości.

Aplikacja jest przeznaczona na konkretnych organizacji, zajmujących się produkcją filmową i telewizyjną - dla producentów filmowych, dla pracowników studia telewizyjnego, reżyserów, operatorów, asystentów operatora. Istnieją jednak wersje aplikacji przeznaczone na rynek masowy, dla użytkowników kamer cyfrowych (livestream, materiały filmowe)

4.4 Analiza SWOT organizacji

Tylko jeśli dla konkretnej organizacji

Wystarczy sama tabela 2x2 (silne-słabe-szanse-zagrożenia)

<p>Silne:</p> <ul style="list-style-type: none">• Możliwość uniezależnienia operatorów filmowych/reżysera od dodatkowych osób poruszających się wokół kamery• Przydatne narzędzie również w trakcie pandemii, kiedy trzeba zachowywać dystans społeczny• Oprogramowanie zwłaszcza jest bardzo użyteczne w studiu telewizyjnym, czyli tworzona jest konkurencja dla ARRI Multicam• Obsługa kamer od wielu producentów i przy podłączeniu ich do stacji bazowej – obsługa podglądu z kilku, oraz możliwości sterowania nimi• Opieranie się na społeczności, co prawdopodobnie może być reklamą przekazywaną pocztą pantoflową	<p>Słabe:</p> <ul style="list-style-type: none">• Filmowcy wolą mieć pełną kontrolę manualną nad swoim projektem/wizją• Prawdopodobnie również będą woleli korzystać z oficjalnych narzędzi producenta• Opieranie się na społeczności
<p>Szanse:</p> <ul style="list-style-type: none">• Możliwość dostrzeżenia narzędzia przez hobbystów• Operatorzy filmowi nie korzystający konkretnie z jednej marki kamery (np. używana jest kamera RED z kamerą od Blackmagic) będą mieli możliwość kontroli nad obiema kamerami jednocześnie• Stworzenie taniego ekosystemu, które będą mogli hobbysci rozwijać zgodnie z własnymi potrzebami	<p>Zagrożenia:</p> <ul style="list-style-type: none">• Zignorowanie narzędzia przez filmowców• Opieranie się na społeczności nie zawsze jest najlepszym rozwiązaniem, kiedy nie wszyscy posiadają odpowiednie zdolności• Narzędzie może nie być tak samo dobre, oraz mieć taką samą funkcjonalność, lub lepszą jak to samo narzędzie pod kamery danego producenta od jego twórców.

5 Specyfikacja wymagań

5.1 Charakterystyka ogólna

5.1.1 Definicja produktu

Zestaw narzędzi do automatyzacji ustawień optycznych kamer filmowych, urządzeń dla operatorów filmowych oraz podglądu na żywo

5.1.2 Podstawowe założenia

Narzędzie jest rozwiązaniem, które umożliwia automatyzację ustawień optycznych kamer cyfrowych i innych urządzeń dla operatorów filmowych. Podstawowymi założeniami jest automatyzacja pewnych procesów technicznych zachodzących na planie, które zwykle realizowane są manualnie. Proces ten ułatwia pracę oraz organizację w czasie tworzenia materiału filmowego osób odpowiedzialnych za nastawy przysłony/ogniskowej/ostrości, co będzie mogło być zastąpione przez automat lub przez asystentów mogących dokonywać zmian ustawień bezprzewodowo.

Narzędzie umożliwia także podgląd na żywo obrazu z kamer w wersji surowej lub z nałożonym lutem. Przy podłączeniu kamery do stacji bazowej za pomocą kabla Ethernet jest możliwa obsługa kilku kamer naraz, wraz z ich podglądem na żywo.

5.1.3 Cel biznesowy

co organizacja docelowa chce osiągnąć wdrażając system

Celem biznesowym jest zaimplementowanie zestawu tych narzędzi, aby zautomatyzować i ułatwić proces produkcji filmowej. Organizacja docelowa będzie mogła dzięki temu zaoszczędzić na kosztach oraz na czasie produkcji

5.1.4 Użytkownicy

Użytkownikami tego systemu są osoby odpowiedzialne za produkcję filmu, m.in.:

- Użytkownik darmowy
- Użytkownik Premium (z wykupioną licencją)
- Użytkownik Premium z wykupioną subskrypcją

5.1.5 Korzyści z systemu

dla poszczególnych grup użytkowników – każdy element z unikalnym numerem identyfikacyjnym

- Użytkownik darmowy – ma dostęp do aplikacji, może udostępniać obraz z kamery na różne usługi, ale jest ograniczony tylko do jednej kamery jednocześnie.
- Użytkownik premium (z wykupioną licencją) – użytkownik, który wykupił licencję ma dostęp do wszystkich możliwych opcji (np. podgląd z wielu kamer jednocześnie), jednak bez aktualizacji, która pojawia się co kilka lat, a jedynie obejmują go łatki bezpieczeństwa i czasami uaktualniane kamery. Licencja umożliwia zakup tylko i wyłącznie danej wersji oprogramowania.

- Użytkownik premium (z wykupioną subskrypcją) – To samo co premium, ale ma do czasu trwania subskrypcji aktualizacje, oraz możliwość synchronizowania ustawień z chmurą.

5.1.6 Ograniczenia projektowe i wdrożeniowe

W przypadku wdrożenia systemu realizacja odbędzie się za pośrednictwem technologii takich jak: Arduino, raspberry PI. Kompatybilność z kamerami cyfrowymi, brak pełnej funkcjonalności dla kamer analogowych (podglądu na żywo). Wszelkie moduły tych narzędzi są synchronizowane w ramach jednego systemu, a więc mogą być zarządzane i kontrolowane zdalnie

5.2 Wymagania funkcjonalne

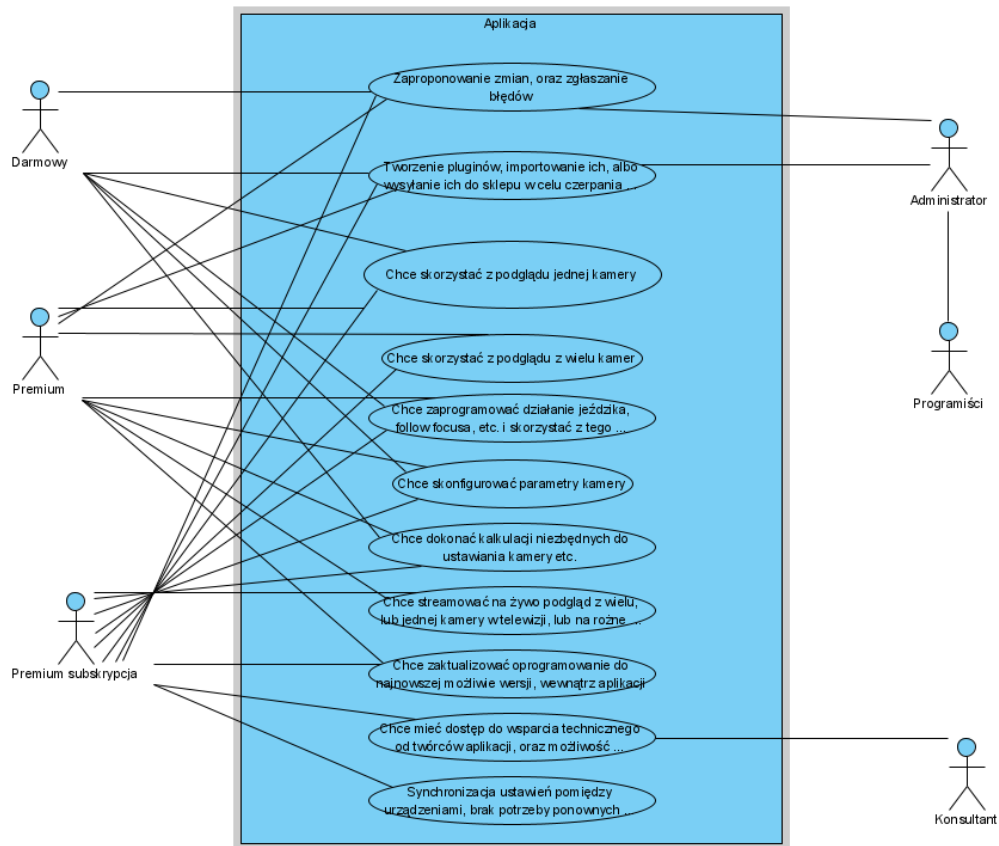
5.2.1 Lista wymagań

- Wsparcie dla wielu producentów kamer
- Przy podłączeniu kamer do stacji bazowej za pomocą kabla ethernetowego, jest możliwość podglądu na żywo z wielu kamer,
- Możliwość konfiguracji ustawień kamery, lub wielu kamer (za pomocą stacji bazowej)
- Wspieranie urządzeń od renomowanych producentów akcesoriów
- Możliwość zaprogramowania zachowania urządzeń (scenariuszy) we własnej wersji interfejsu „Scratchowego”.
- Możliwość tworzenia własnych narzędzi typu follow Focus, jeździk i sparowania go z programem
- Możliwość streamowania obrazu z kamery na żywo do telewizji, lub różnych internetowych serwisów (typu YouTube, Twitch etc.)
- Synchronizacja ustawień do chmury przy wykupieniu odpowiedniej subskrypcji przez klienta
- Możliwość nakładania LUTów na podglądany obraz z kamery
- Narzędzie posiada listę zalecanych pluginów, oraz wbudowany sklep z nimi, wraz z opiniami innych użytkowników.
- Możliwość importu pluginów z jakichkolwiek innych źródeł za pomocą pliku, jak i jego eksportu z programu.
- Możliwość przesłania własnego pluginu do weryfikacji przez twórców/administrację pod kątem awaryjności/działania/braku zagrożeń i możliwość wrzucenia tego do sklepu, oraz czerpania z tego korzyści (70% dla twórców pluginu, 30% dla twórców aplikacji)
- Możliwość nakładania na nagrywany materiał informacji potrzebnych dla montażysty (scena, ujęcie, etc.)
- Wbudowane narzędzia do kalkulacji crop factor matrycy przy wybraniu odpowiedniej rozdzielczości/proporcji obrazu, kalkulacji czasu

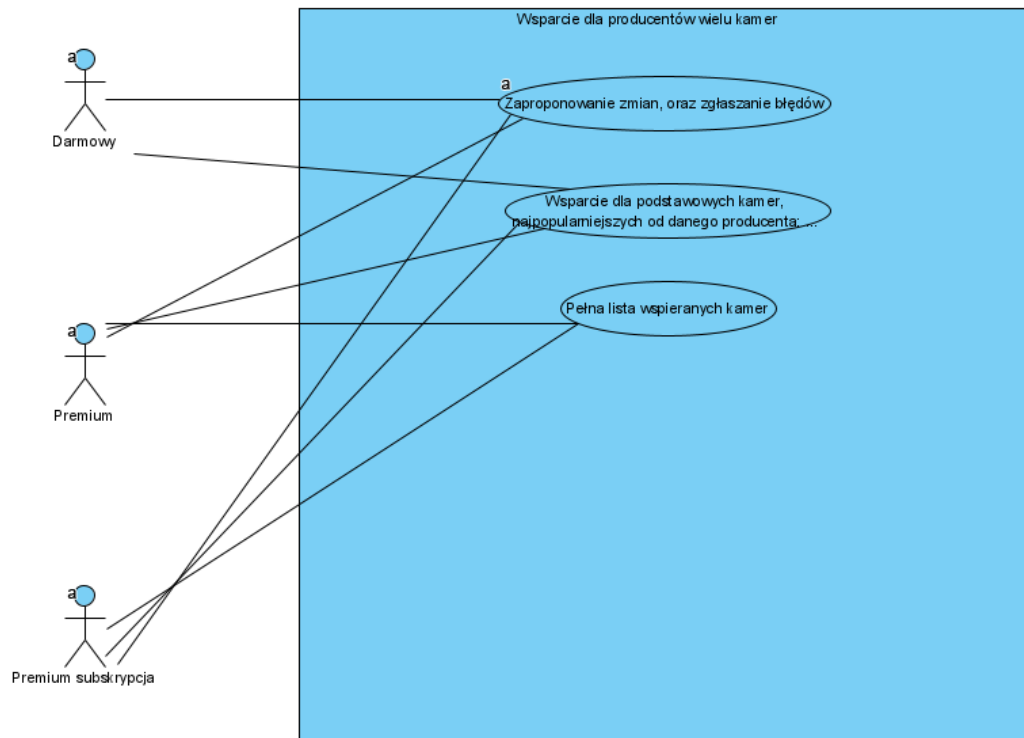
migawki/shutter angle (aby uzyskać obraz bez migotania w danej części świata (60Hz w PA, 50Hz w Europie i przynajmniej większej części Azji), kalkulacji ekspozycji (parametry wejściowe: shutter angle, filtr nd, frame rate, transmisja światła obiektywu)

5.2.2 Diagramy przypadków użycia

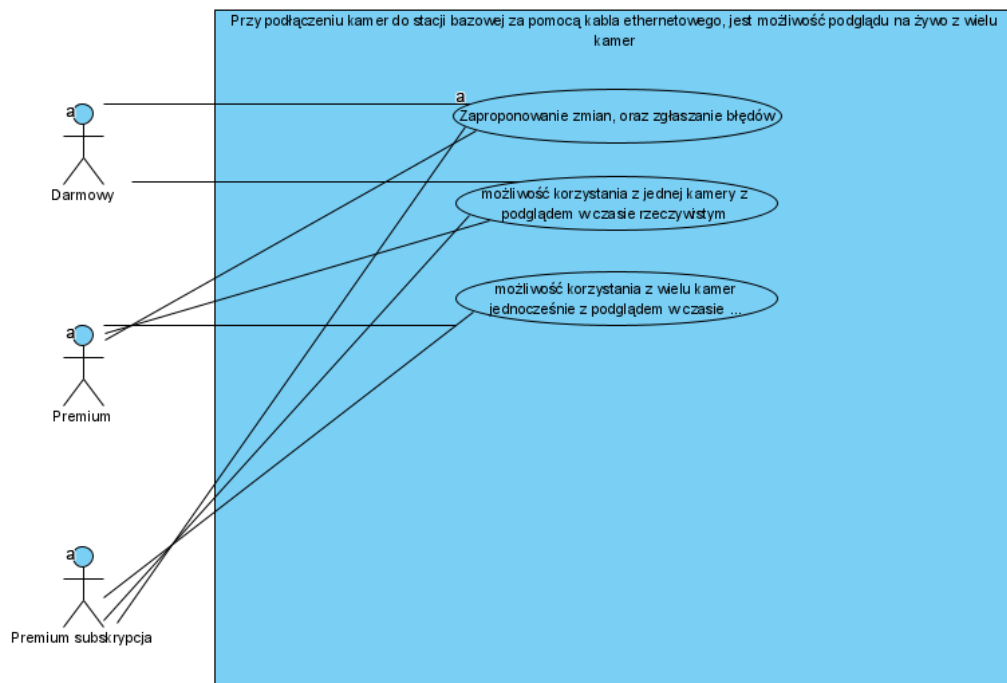
Tutaj same diagramy – bez specyfikacji, ale każdy diagram z tytułem i na osobnej stronie



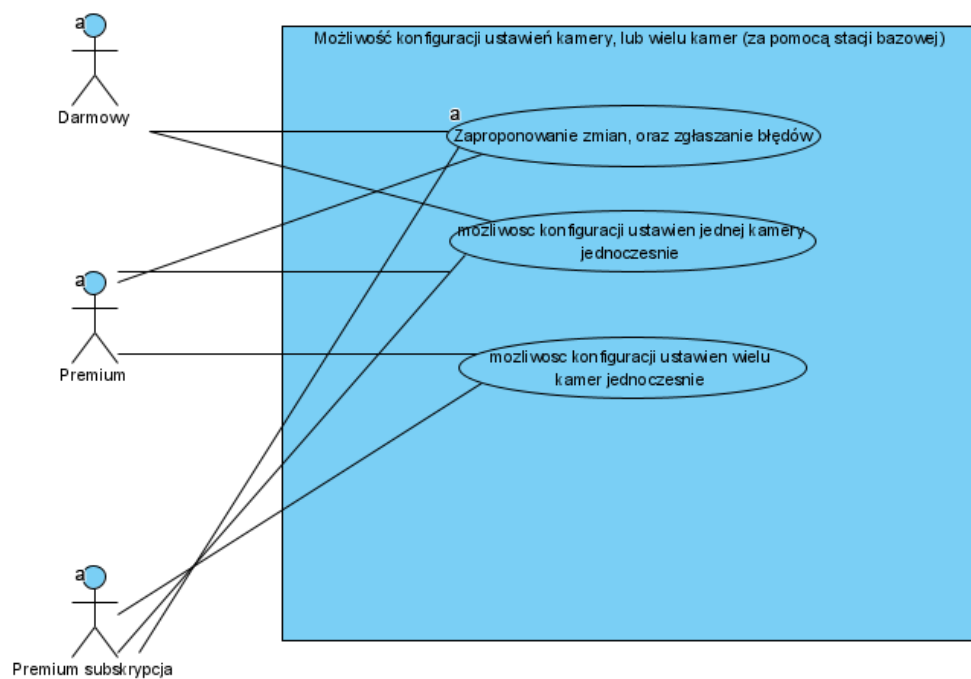
1. Wsparcie dla producentów wielu kamer



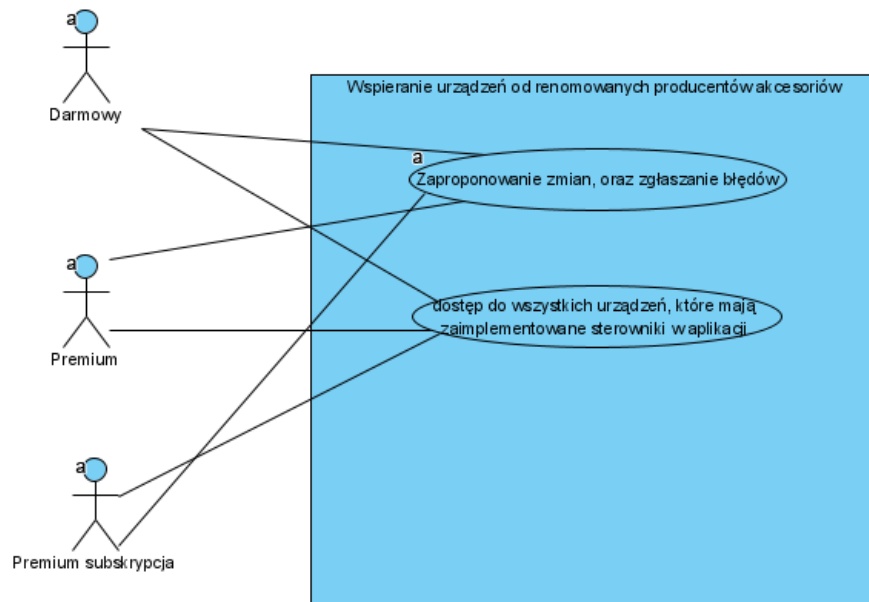
2. Przy podłączeniu kamer do stacji bazowej za pomocą kabla ethernetowego, jest możliwość podglądu na żywo z wielu kamer



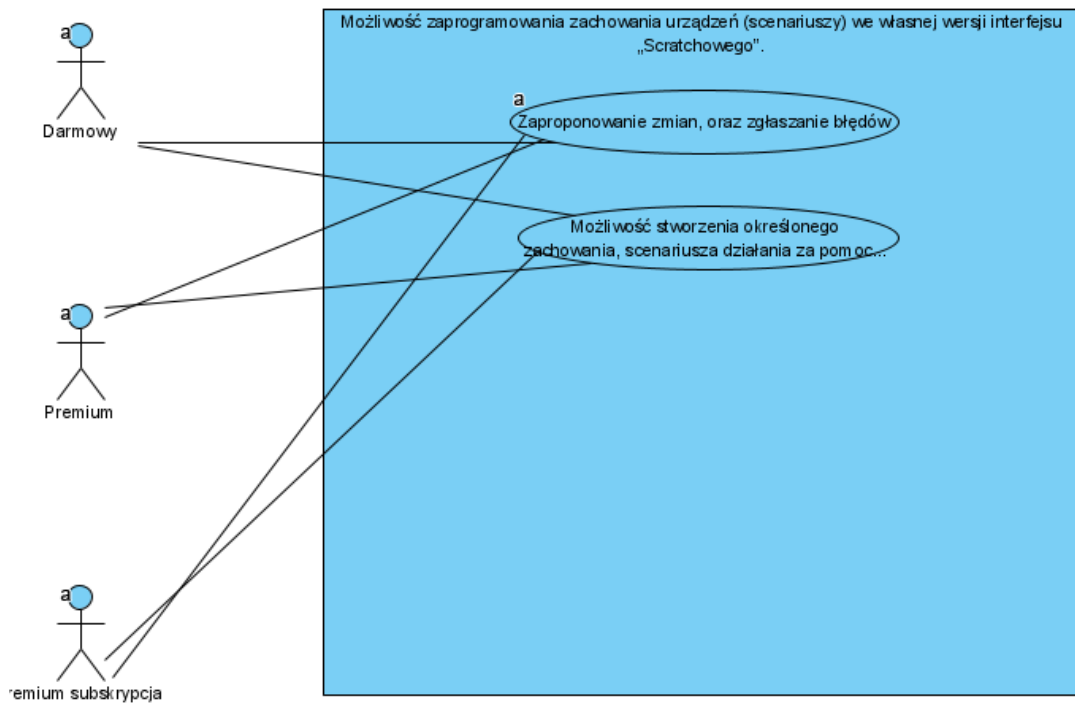
1. Możliwość konfiguracji ustawień kamery, lub wielu kamer (za pomocą stacji bazowej)



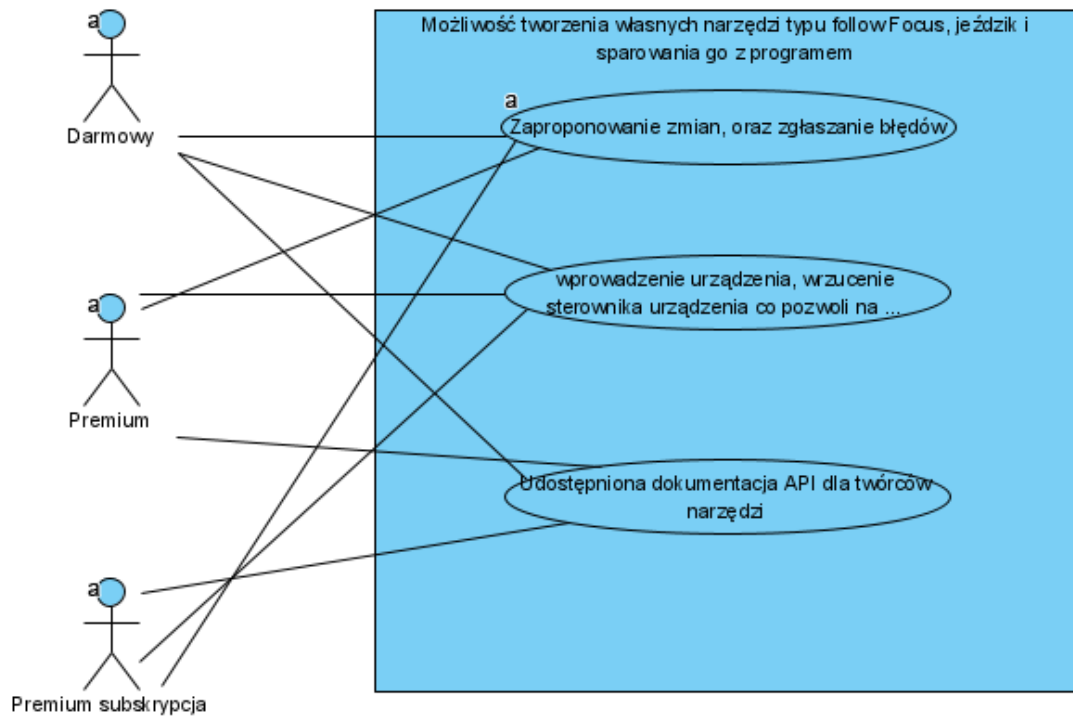
2. Wsparanie urządzeń od renomowanych producentów akcesoriów



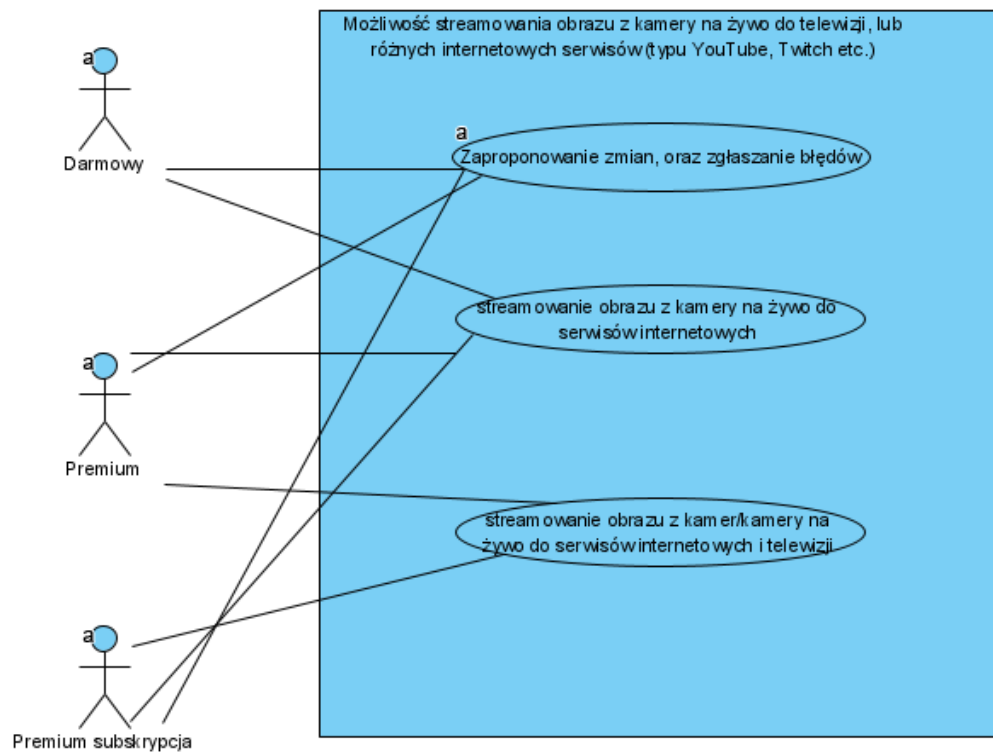
3. Możliwość zaprogramowania zachowania urządzeń (scenariuszy) we własnej wersji interfejsu „Scratchowego”.



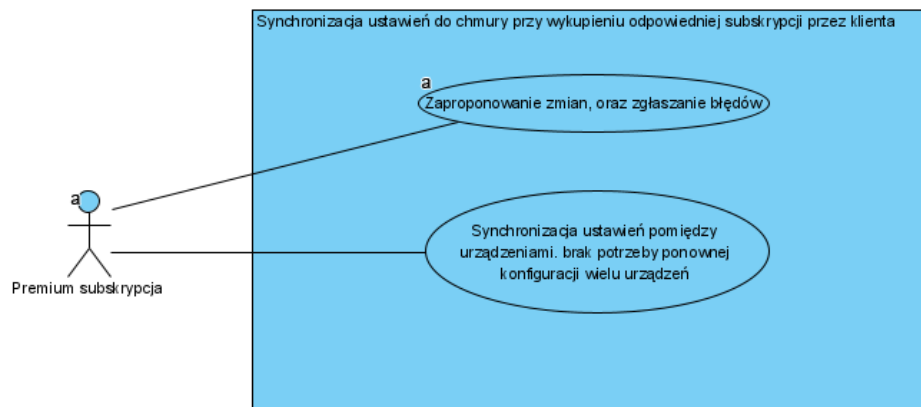
4. Możliwość tworzenia własnych narzędzi typu follow Focus, jeździak i sparowania go z programem



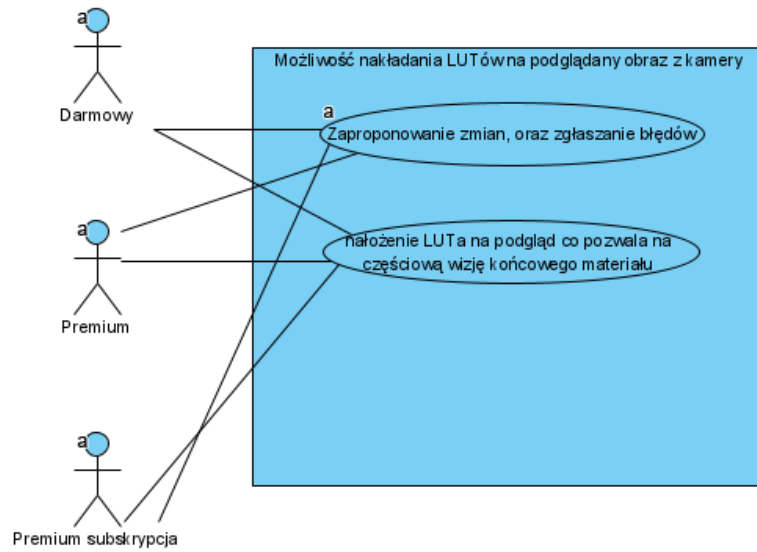
5. Możliwość streamowania obrazu z kamery na żywo do telewizji, lub różnych internetowych serwisów (typu YouTube, Twitch etc.)



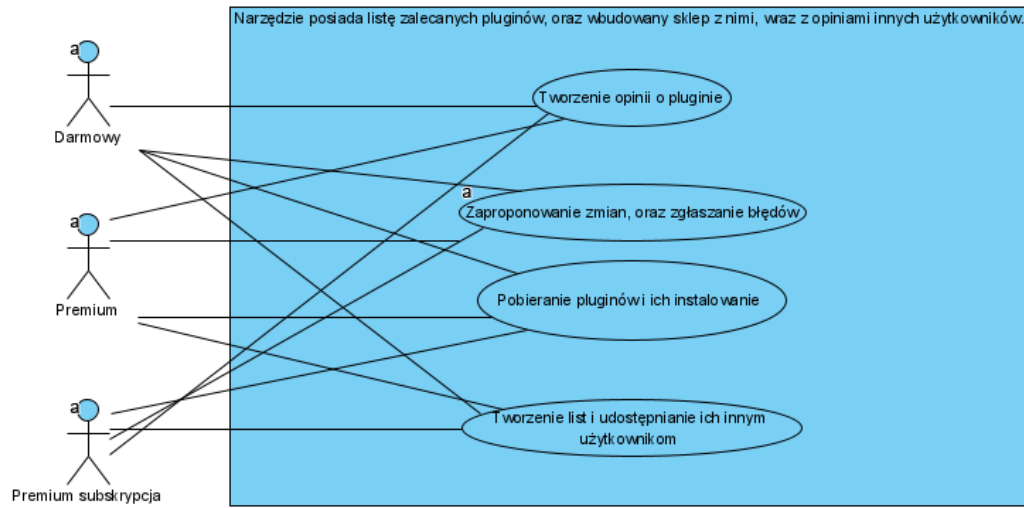
6. Synchronizacja ustawień do chmury przy wykupieniu odpowiedniej subskrypcji przez klienta



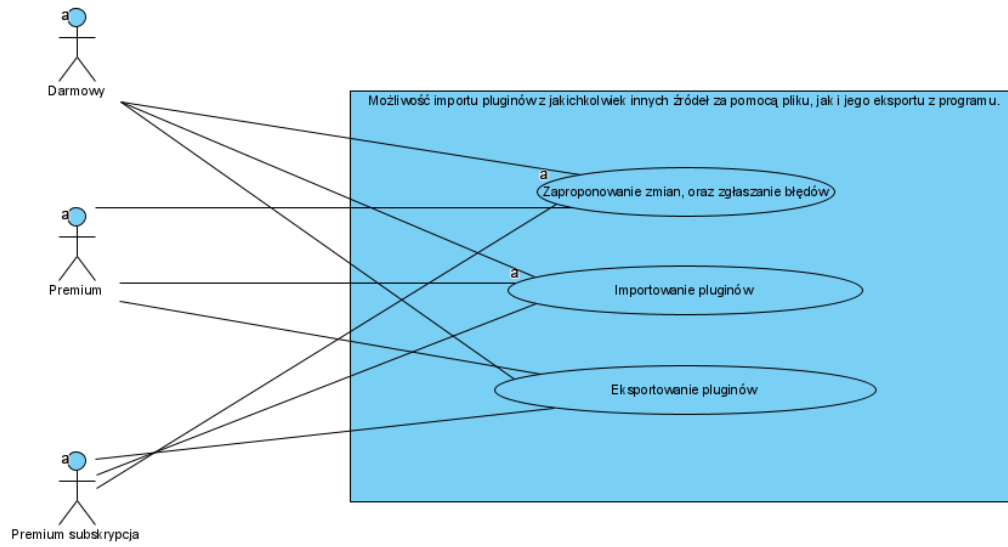
7. Możliwość nakładania LUTów na podglądany obraz z kamery



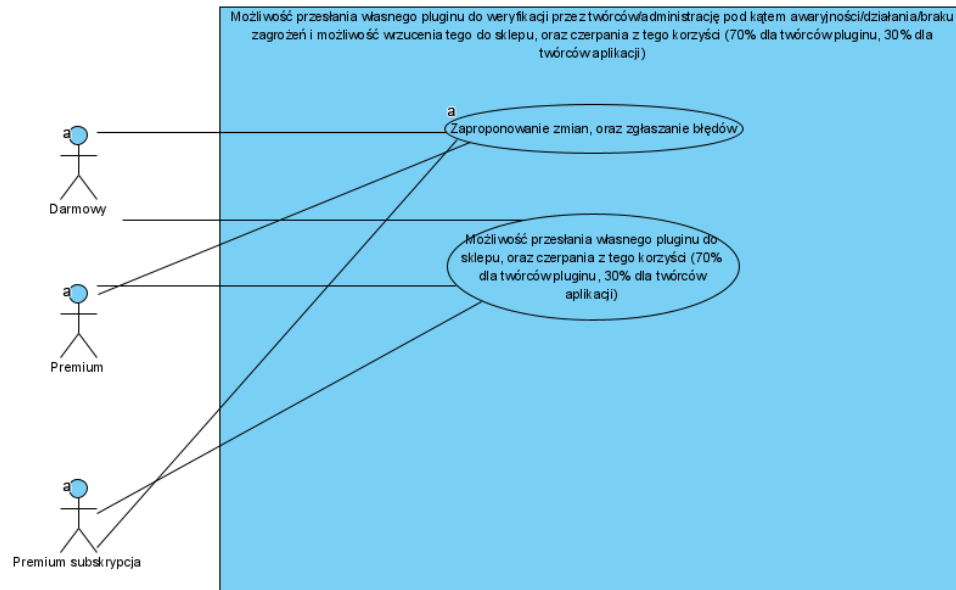
8. Narzędzie posiada listę zalecanych pluginów, oraz wbudowany sklep z nimi, wraz z opiniami innych użytkowników.



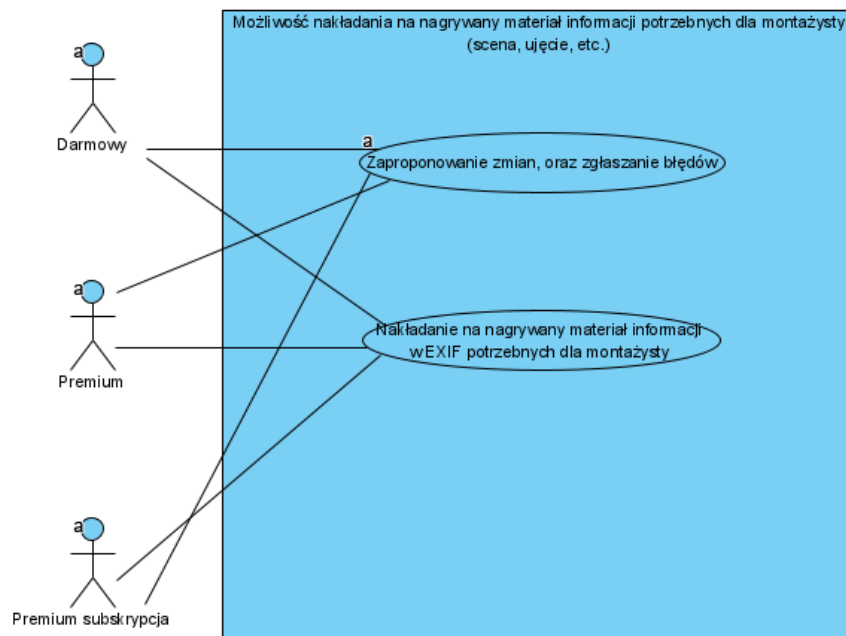
9. Możliwość importu pluginów z jakichkolwiek innych źródeł za pomocą pliku, jak i jego eksportu z programu.



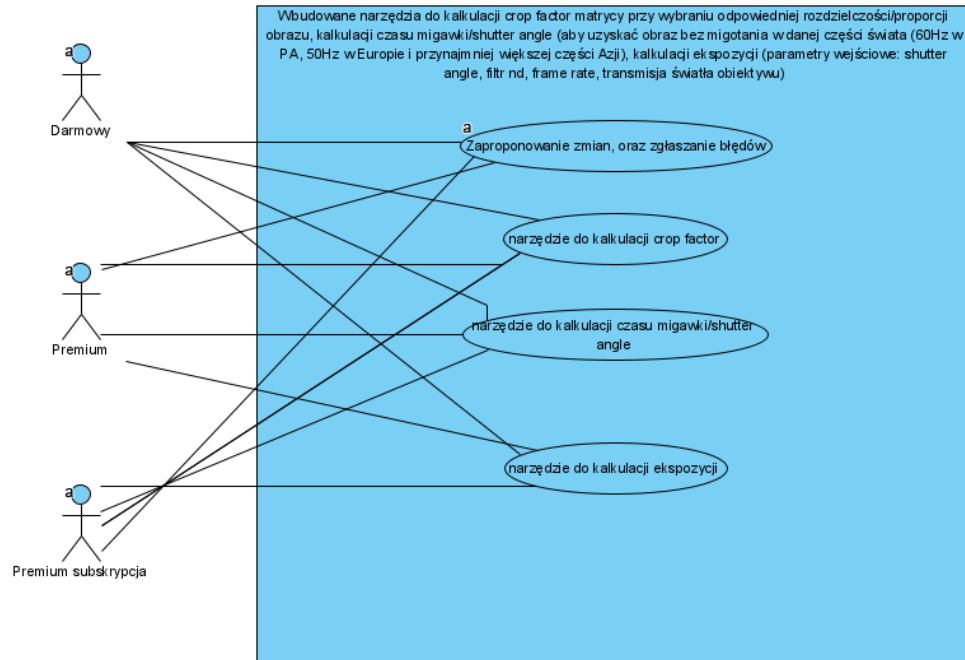
10. Możliwość przesłania własnego pluginu do weryfikacji przez twórców/administrację pod kątem awaryjności/działania/braku zagrożeń i możliwość wrzucenia tego do sklepu, oraz czerpania z tego korzyści (70% dla twórców pluginu, 30% dla twórców aplikacji)



11. Możliwość nakładania na nagrywany materiał informacji potrzebnych dla montażysty (scena, ujęcie, etc.)



12. Wbudowane narzędzia do kalkulacji crop factor matrycy przy wybraniu odpowiedniej rozdzielczości/proporcji obrazu, kalkulacji czasu migawki/shutter angle (aby uzyskać obraz bez migotania w danej części świata (60Hz w PA, 50Hz w Europie i przynajmniej większej części Azji), kalkulacji ekspozycji (parametry wejściowe: shutter angle, filtr nd, frame rate, transmisja światła obiektywu)



5.2.3 Szczegółowy opis wymagań

każde na nowej stronie wg następujących punktów:

- Numer – jako ID
- Nazwa
- Uzasadnienie biznesowe – odwołanie (-a) do elementów wymienionych w 5.1.5. (id i treść elementu, do którego się odwołujemy)
- Użytkownicy
- Scenariusze, dla każdego z nich:
 - Warunki początkowe
 - **Przebieg działań** – numerowana lista kroków, ze wskazaniem, kto realizuje dany krok
 - Efekty – warunki końcowe
 - Wymagania niefunkcjonalne – szczegółowe wobec poszczególnych wymagań funkcjonalnych
 - Częstotliwość - na skali 1-5 lub BN-BW
 - Istotność – inaczej: zależność krytyczna, znaczenie - na skali 1-5 lub BN-BW

- 1
- Wsparcie dla producentów wielu kamer
- Uzasadnienie biznesowe – korzyści dla użytkowników, którzy korzystają z wielu kamer od różnych producentów, oraz potencjalna możliwość zainteresowania ich tym programem. Najważniejsze założenie, bo bez niego program nie ma sensu istnienia.
- Subskrybenci, Posiadacze pojedynczej licencji, darmowi (bardzo ograniczona lista wspieranych kamer)
- Scenariusz:
 - Warunki początkowe: Udostępnienie API swoich kamer przez producenta
 - **Przebieg działań:**
 - .1. Programiści implementują API, oraz funkcjonalności konieczne do obsługi kamery**
 - .2. Użytkownik podłącza kamerę do komputera i uruchamia aplikację**
 - .3. Użytkownik korzysta z kamery**
 - .4. W przypadku niewspieranej kamery – zgłoszenie niewspieranej kamery do administratora, który przekazuje informację dalej**
 - Efekty – Zadowolony użytkownik z aplikacji, która wspiera jego kamerę
 - Wymagania niefunkcjonalne – wydajność, bezpieczeństwo, poprawność, adaptowalność, dostępność, użyteczność.
 - Częstotliwość - 5
 - Istotność – 5

- 2
- Przy podłączeniu kamer do stacji bazowej za pomocą kabla ethernetowego, jest możliwość podglądu na żywo z wielu kamer
- Uzasadnienie biznesowe – korzyści dla użytkowników, którzy korzystają z wielu kamer od różnych producentów, oraz potencjalna możliwość zainteresowania ich tym programem. Najważniejsze założenie, ale tylko dla użytkowników, którzy zapłacili za aplikację.
- Subskrybenci, Posiadacze pojedynczej licencji (z bazą wspieranych, bez możliwości aktualizacji), darmowi (brak uprawnień do tej opcji)
- Scenariusz:
 - Warunki początkowe: Posiadanie stacji bazowej, podłączonych kamer do tej stacji bazowej za pomocą kabla ethernetowego
 - **Przebieg działań:**
 - .1. Użytkownik podłącza kamery do stacji bazowej**
 - .2. Uruchamia aplikację**
 - .3. Użytkownik ma możliwość korzystania z wielu kamer naraz i możliwości ich podglądu**
 - Efekty – Możliwość podglądu z wielu kamer naraz
 - Wymagania niefunkcjonalne – wydajność, bezpieczeństwo, poprawność, adaptowalność, dostępność, użyteczność.
 - Częstotliwość - 5
 - Istotność – 5

- 3
- Możliwość konfiguracji ustawień kamery, lub wielu kamer (za pomocą stacji bazowej)
- Uzasadnienie biznesowe – korzyści dla użytkowników, którzy korzystają z wielu kamer od różnych producentów, oraz potencjalna możliwość zainteresowania ich tym programem. Najważniejsze założenie, bo bez niego program nie ma sensu istnienia.
- Subskrybenci, Posiadacze pojedynczej licencji (z bazą wspieranych, bez możliwości aktualizacji), darmowi (możliwość konfiguracji ustawień tylko pojedynczej kamery)
- Scenariusze, dla każdego z nich:
 - Warunki początkowe: Posiadanie kamery podłączonej do komputera/stacji bazowej
 - **Przebieg działań:**
 - .1. Użytkownik podłącza kamerę**
 - .2. Uruchamia aplikację**
 - .3. Użytkownik korzysta z podglądu**
 - .4. Użytkownik zmienia parametry kamery z możliwością podglądu zmian na bieżąco**
 - Efekty – zmienione parametry kamery
 - Wymagania niefunkcjonalne – wydajność, bezpieczeństwo, poprawność, adaptowalność, dostępność, użyteczność, łatwość konserwacji.
 - Częstotliwość - 5
 - Istotność – 5

- 4
- Wspieranie urządzeń od renomowanych producentów akcesoriów
- Uzasadnienie biznesowe – korzyści dla użytkowników, którzy korzystają z akcesoriów od różnych producentów, oraz potencjalna możliwość zainteresowania ich tym programem.
- Subskrybenci, Posiadacze pojedynczej licencji (z bazą wspieranych, bez możliwości aktualizacji), darmowi (bardzo ograniczona lista wspieranych akcesoriów)
- Scenariusze, dla każdego z nich:
 - Warunki początkowe: Posiadanie akcesorium
 - **Przebieg działań:**
 - .1. Podpinanie akcesorium**
 - .2. Parowanie go z programem**
 - .3. Korzystanie z akcesorium**
 - Efekty – Możliwość korzystania z akcesorium
 - Wymagania niefunkcjonalne – wydajność, bezpieczeństwo, poprawność, adaptowalność, dostępność, użyteczność.
 - Częstotliwość - 5
 - Istotność – 4

- 5
- Możliwość zaprogramowania zachowania urządzeń (scenariuszy) we własnej wersji interfejsu „Scratchowego”.
- Uzasadnienie biznesowe – potencjalna możliwość zainteresowania ich tym programem. Najważniejsze założenie, bo bez niego program nie ma sensu istnienia.
- Subskrybenci, Posiadacze pojedynczej licencji (z bazą wspieranych, bez możliwości aktualizacji), darmowi (bardzo ograniczona lista wspieranych kamer)
- Scenariusz:
 - Warunki początkowe: Posiadanie kilku akcesoriów, oraz kamery sparowanej z komputerem, oraz poprawnie skalibrowanej
 - **Przebieg działań:**
 - .1. Wcześniej bardzo dokładnie skalibrowane urządzenie zostaje dodane do możliwości oskryptowania**
 - .2. Interfejs scratchowy oferuje zaimplementowanie wielu urządzeń, zaprogramowanych ruchów w kilku osiach, obrotów etc.**
 - .3. Po zaimplementowaniu skryptu można go wykorzystać przy kręceniu materiału, lub żeby przetestować**
 - Efekty – użytkownik może korzystać z zaprogramowanych ruchów
 - Wymagania niefunkcjonalne – wydajność, poprawność, adaptowalność, dostępność, użyteczność.
 - Częstotliwość - 4
 - Istotność – 5

- 6
- Możliwość tworzenia własnych narzędzi typu follow Focus, jeźdźnik i sparowania go z programem.
- Uzasadnienie biznesowe – korzyści dla użytkowników, którzy korzystają z wielu kamer od różnych producentów, oraz potencjalna możliwość zainteresowania ich tym programem. Najważniejsze założenie, bo bez niego program nie ma sensu istnienia.
- Subskrybenci, Posiadacze pojedynczej licencji (z bazą wspieranych, bez możliwości aktualizacji), darmowi (bardzo ograniczona lista wspieranych kamer)
- Scenariusze, dla każdego z nich:
 - Warunki początkowe: Zapoznanie się użytkownika z API, oraz możliwościami programu, stworzenie oraz podłączenie poprawnie działającego własnego akcesorium.
 - **Przebieg działań:**
 - .1. Po podłączeniu urządzenia, należy wejść do menu urządzeń podłączonych do komputera, oraz go wybrać, oraz przypisać mu kategorię.**
 - .2. Po wybraniu kategorii należy skalibrować urządzenie, program będzie wykonywać „własne ruchy” i użytkownik, będzie wprowadzał co było efektem działania komputera.**
 - .3. Po poprawnej kalibracji można zacząć już korzystać z własnego rozwiązania.**
 - Efekty – użytkownik może implementować własne rozwiązania i z nich korzystać
 - Wymagania niefunkcjonalne – poprawność, adaptowalność, dostępność, użyteczność.
 - Częstotliwość - 2
 - Istotność – 5

- 7
- Możliwość streamowania obrazu z kamery na żywo do telewizji, lub różnych internetowych serwisów (typu YouTube, Twitch etc.)
- Uzasadnienie biznesowe –potencjalna możliwość zainteresowania ich tym programem. Najważniejsze założenie, bo bez niego program nie ma sensu istnienia.
- Subskrybenci, Posiadacze pojedynczej licencji (z bazą wspieranych, bez możliwości aktualizacji), darmowi (bardzo ograniczona lista wspieranych kamer)
- Scenariusze, dla każdego z nich:
 - Warunki początkowe: Podłączona kamera do komputera i włączona aplikacja.
 - **Przebieg działań:**
 - .1. Użytkownik wybiera opcję wewnątrz aplikacji, na który serwis chce się udostępnić obraz.**
 - .2. Użytkownik wprowadza dane logowania do serwisu**
 - .3. Użytkownik po zalogowaniu się do danego serwisu, może wybrać ustawienia kamery, które pomogą mu dostosować obraz.**
 - .4. Użytkownik może nakładkę na nagrania.**
 - .5. Użytkownik klika przycisk „LIVE”, aby udostępnić obraz z kamery.**
 - Efekty – Obraz z kamery jest przesyłany do wybranego serwisu, wybranego przez użytkownika
 - Wymagania нефункционалне – wydajność, bezpieczeństwo, poprawność, adaptowalność, użyteczność, elastyczność, dostępność.
 - Częstotliwość - 2
 - Istotność – 5

- 8
- Synchronizacja ustawień do chmury przy wykupieniu odpowiedniej subskrypcji przez klienta
- Uzasadnienie biznesowe – pasywny dochód, który pozwala na utrzymanie serwerów. Potencjalna możliwość zainteresowania użytkowników subskrypcją.
- Subskrybenci
- Scenariusze, dla każdego z nich:
 - Warunki początkowe – wykupiona subskrypcja, zalogowane konto użytkownika, połączone z internetem
 - **Przebieg działań:**
 - .1. Użytkownik dokonuje jakichkolwiek zmian w programie.**
 - .2. Program wysyła do chmury obecne ustawienia, oraz przechowuje kilka poprzednich kopii zapasowych na wypadek zepsucia.**
 - .3. Jeżeli program jest na nowo zainstalowany: wyszukuje ostatnie kopie zapasowe i pobiera najnowszą w tle.**
 - Efekty – użytkownik nie musi nic na nowo konfigurować, ponieważ ma zapis wszystkich zmian jakich dokonał w ustawieniach w chmurze.
 - Wymagania niefunkcjonalne – bezpieczeństwo, zabezpieczenia, poprawność, adaptowalność, dostępność, użyteczność, przenośność.
 - Częstotliwość - 3
 - Istotność – 3

- 9
- Możliwość nakładania LUTów na podglądany obraz z kamery
- Uzasadnienie biznesowe – korzyści dla użytkowników, którzy korzystają z wielu kamer od różnych producentów, oraz potencjalna możliwość zainteresowania ich tym programem.
- Subskrybenci, Posiadacze pojedynczej licencji, darmowi
- Scenariusze, dla każdego z nich:
 - Warunki początkowe: podłączona kamera z obrazem wyświetlanym na żywo.
 - **Przebieg działań:**
 - .1. Użytkownik wybiera z listy zaimportowanych/wbudowanych LUTów, taki który chciałby nanieść na obraz z kamery.**
 - .2. Aby wybrać LUT należy kliknąć prawym przyciskiem myszy LUT i wybrać opcję „nałóż”**
 - Efekty – użytkownik może widzieć na żywo częściową wizję końcowego materiału.
 - Wymagania niefunkcjonalne – wydajność, poprawność, adaptowalność, dostępność, użyteczność.
 - Częstotliwość - 1
 - Istotność – 2

- 10
- Narzędzie posiada listę zalecanych pluginów, oraz wbudowany sklep z nimi, wraz z opiniami innych użytkowników.
- Uzasadnienie biznesowe – korzyści dla użytkowników, oraz potencjalna możliwość zainteresowania ich tym programem. Będzie kojarzyło się użytkownikom z dużym wsparciem społeczności, co z pewnością zainteresuje przyszłych potencjalnych użytkowników.
- Subskrybenci, Posiadacze pojedynczej licencji, darmowi
- Scenariusze, dla każdego z nich:
 - Warunki początkowe – włączona aplikacja, połączenie z internetem, zalogowane konto użytkownika.
 - **Przebieg działań:**
 - .1. Użytkownik otwiera wbudowany sklep, otwiera listę zalecanych pluginów.**
 - .2. Użytkownik pobiera plugin i go aktywuje w głównej części programu.**
 - .3. Jeżeli plugin będzie dobry, to użytkownik może pochwalić autorów, a jeżeli zły to może zostawić konstruktywną, jak i mniej krytykę.**
 - Efekty – użytkownik ma wiele pluginów do wyboru, które pomogą mu przy tworzeniu filmów, oraz może wyrażać swoją opinię na ich temat w sklepie.
 - Wymagania niefunkcjonalne – wydajność, bezpieczeństwo, poprawność, adaptowalność, dostępność, użyteczność.
 - Częstotliwość - 4
 - Istotność – 4

- 11
- Możliwość importu pluginów z jakichkolwiek innych źródeł za pomocą pliku, jak i jego eksportu z programu.
- Uzasadnienie biznesowe – korzyści dla użytkowników, oraz potencjalna możliwość zainteresowania ich tym programem. Będzie kojarzyło się użytkownikom z dużym wsparciem społeczności, co z pewnością zainteresuje przyszłych potencjalnych użytkowników.
- Subskrybenci, Posiadacze pojedynczej licencji, darmowi
- Scenariusze, dla każdego z nich:
 - Warunki początkowe – włączona aplikacja.
 - **Przebieg działań:**
 - .1. Użytkownik otwiera listę pluginów, wybiera opcje importu pluginów**
 - .2. Użytkownik importuje plugin i go aktywuje w głównej części programu.**
 - .3. Jeżeli plugin będzie dobry, to użytkownik może podzielić się nim eksportując go (jeżeli nie był płatny w sklepie).**
 - Efekty – użytkownik ma wiele pluginów do wyboru, które pomogą mu przy tworzeniu filmów.
 - Wymagania niefunkcjonalne – wydajność, bezpieczeństwo, poprawność, adaptowalność, dostępność, użyteczność.
 - Częstotliwość - 2
 - Istotność – 4

- 12
- Możliwość przesłania własnego pluginu do weryfikacji przez twórców/administrację pod kątem awaryjności/działania/braku zagrożeń i możliwość wrzucenia tego do sklepu, oraz czerpania z tego korzyści (70% dla twórców pluginu, 30% dla twórców aplikacji)
- Uzasadnienie biznesowe – korzyści dla użytkowników, czerpanie z tego korzyści (70% dla twórców pluginu, 30% dla twórców aplikacji), oraz potencjalna możliwość zainteresowania ich tym programem.
- Subskrybenci, Posiadacze pojedynczej licencji, darmowi
- Scenariusze, dla każdego z nich:
 - Warunki początkowe – włączona aplikacja, połączenie z internetem, zweryfikowane konto, stworzony plugin.
 - **Przebieg działań:**
 - .1. Użytkownik przesyła swój plugin do weryfikacji, ustala kwotę jaką chce otrzymywać za swoją pracę (za darmo/płatne)**
 - .2. Administratorzy weryfikują w piaskownicy plugin, jeżeli nie ma żadnych zagrożeń to przekazywany jest do ostatniego etapu.**
 - .3. Plugin pojawia się w sklepie, inni użytkownicy mogą go pobierać, a twórca może czerpać z tego korzyści.**
 - Efekty – inni użytkownicy mają wiele pluginów do wyboru, które pomogą im przy tworzeniu filmów, a twórca może czerpać z tego korzyści.
 - Wymagania niefunkcjonalne – wydajność, bezpieczeństwo, zabezpieczenia, poprawność, adaptowalność, dostępność, użyteczność.
 - Częstotliwość - 5
 - Istotność – 4

- 13
- Możliwość nakładania na nagrywany materiał informacji potrzebnych dla montażysty (scena, ujęcie, etc.)
- Uzasadnienie biznesowe – potencjalna możliwość zainteresowania tym programem.
- Subskrybenci, Posiadacze pojedynczej licencji, darmowi
- Scenariusze, dla każdego z nich:
 - Warunki początkowe – dostęp do nagranych/nagrywanego materiału, uruchomiona aplikacja.
 - **Przebieg działań:**
 - .1. Użytkownik wybiera opcje EXIF/DATA przy widoku wybranego materiału.**
 - .2. Użytkownik może nanieść informację na klip potrzebnych na rzecz post produkcji.**
 - Efekty – klip ma naniesione informacje, które są potrzebne montażysty i innym.
 - Wymagania niefunkcjonalne – wydajność, bezpieczeństwo, poprawność, adaptowalność, dostępność, użyteczność, elastyczność.
 - Częstotliwość - 1
 - Istotność – 2

- 14
- Wbudowane narzędzia do kalkulacji crop factor matrycy przy wybraniu odpowiedniej rozdzielczości/proporcji obrazu, kalkulacji czasu migawki/shutter angle (aby uzyskać obraz bez migotania w danej części świata (60Hz w PA, 50Hz w Europie i przynajmniej większej części Azji), kalkulacji ekspozycji (parametry wejściowe: shutter angle, filtr nd, frame rate, transmisja światła obiektywu)
- Uzasadnienie biznesowe – potencjalna możliwość zainteresowania programem.
- Subskrybenci, Posiadacze pojedynczej licencji, darmowi
- Scenariusze, dla każdego z nich:
 - Warunki początkowe – uruchomiona aplikacja
 - **Przebieg działań:**
 - .1. Użytkownik wybiera, które działanie chce wykonać np. przeliczyć crop factor matrycy przy wybraniu innych proporcji ekranu. Aplikacja sama przelicza jaką rozdzielczość otrzymamy, crop factor, oraz inne informacje.**
 - Efekty – użytkownik może dokonać obliczeń, aby zapewnić odpowiedni kąt widzenia, przekalkulować czas migawki, ekspozycję, aby obraz był zgodny z wizją i bez wad.
 - Wymagania нефunkcjonalne – poprawność, adaptowalność, dostępność, użyteczność.
 - Częstotliwość - 1
 - Istotność – 1

5.3 Wymagania niefunkcjonalne

wobec całego systemu

1. Wydajność – w odniesieniu do konkretnych sytuacji – funkcji systemu
 - Aplikacja musi zapewnić skalowalność i możliwość skalowania na wykorzystanym do tego sprzęcie komputerowym,
 - Aplikacja posiada krótki czas reakcji ze strony użytkowników systemu oraz swoich odrębnych modułów,
 - Aplikacja zapewnia wysoką wydajność w stosunku do zużytej energii i zasobów (zużycie procesora, pamięci RAM, dysku twardego),
 - Aplikacja będzie umożliwiać korzystanie z niej przez 5 użytkowników jednocześnie. Osoby te będą mieć dostęp do infrastruktury z dowolnego miejsca poprzez połączenie internetowe,
 - Stabilność - ciągła, nieprzerwana praca bez względu na liczbę użytkowników i sposób z niej korzystania,
 - Niezawodność - niska podatność na awarie systemu,
 - Utrzymanie wydajności pod dużym obciążeniem systemu.
 - Możliwość obsłużenia do 5 kamer jednocześnie,
 - Struktura aplikacji wraz z danymi nie przekroczy rozmiaru 10 GB.
2. Bezpieczeństwo - utrata, zniszczenie danych, zniszczenie innego systemu przez nasz – wraz z działaniami zapobiegawczymi i ograniczającymi skutki
 - Bezpieczeństwo przed atakami Denial of Service (DOS),
 - Blokowanie adresów IP, z których następuję częsta prośba o połączenie.
 - Komunikacja z aplikacją tylko uprawnionych użytkowników, Autoryzacja użytkowników,
 - Tworzenie regularnych kopii zapasowych danych (ustawienia konfiguracyjne, dane o infrastrukturze systemu ...),
 - Możliwość przywrócenia całego systemu wraz z danymi poprzez zastosowanie punktów przywracania,
 - Bezpieczeństwo fizyczne. Realizowane na poziomie uprawnień dostępu do plików.
 - Możliwość zaszyfrowania danych systemu,
 - Zastosowanie infrastruktury sprzętowej odpornej na uszkodzenia fizyczne, pracę w warunkach terenowych,
 - Synchronizacja danych w chmurze.
3. Zabezpieczenia
 - Implementacja zabezpieczeń przed atakami typu ransomware,
 - Monitorowanie wszelkich nowych połączeń od urządzeń wejścia/wyjścia,
 - Regularne aktualizacje oprogramowania,
 - Zabezpieczenia przed kradzieżą, weryfikacja użytkowników, możliwość zablokowania użyteczności fizycznych urządzeń w przypadku kradzieży.
4. Inne cechy jakości – najlepiej **ilościowo**, żeby można było **zweryfikować (zmierzyć)** – adaptowalność, dostępność, poprawność, elastyczność, łatwość konserwacji, przenośność, awaryjność, testowalność, użyteczność

- Odporność systemu - krótki czas restartu po awarii systemu, znikome prawdopodobieństwo zniszczenia danych w przypadku awarii,
- Aplikacja będzie dostępna dla wszystkich użytkowników po wdrożeniu, bez ograniczeń limitu czasowego,
- Wszystkie błędy krytyczne aplikacji zostaną skutecznie naprawione w ciągu 48 h roboczych,
- Wszyscy użytkownicy mogą zgłaszać błędy na dedykowany adres email i dedykowany numer telefonu,
- Wszystkie błędy aplikacji będą monitorowane. Monitorowany będzie sposób i czas ich naprawy,
- Do aplikacji zgłaszania błędów będą miały dostęp wszystkie osoby ze strony zamawiającego,
- Aplikacja zostanie wdrożona w ciągu maksymalnie 30 dni roboczych,
- Proces wdrożenia aplikacji nie może spowodować braku dostępu do istniejącej infrastruktury sprzętowej i systemowej
- Firma dostarcza: specyfikację sprzętową, specyfikację funkcjonalną, projekt architektury informatycznej

6 Zarządzanie projektem

6.1 Zasoby ludzkie

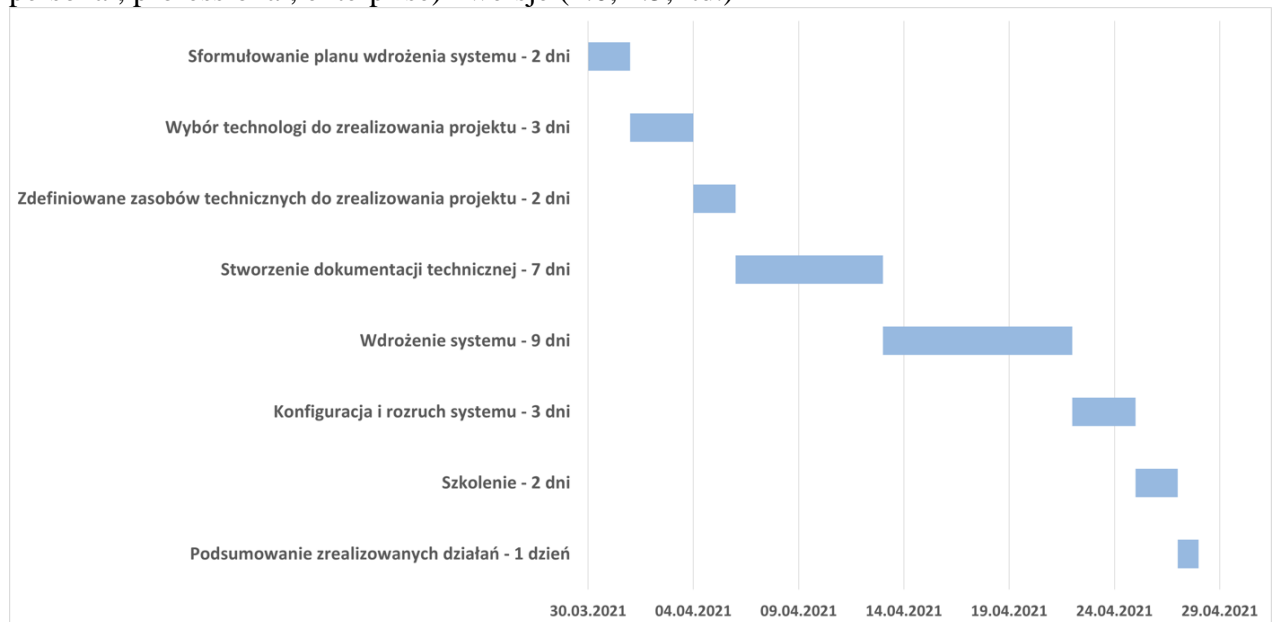
1. Programiści – 15 osób
2. Testerzy – 3 osoby, oraz program „beta” pozwalający na darmowe pozyskanie testerów
3. Graficy – 6 osób
4. Administratorzy sieci – 4 osoby
5. PR – 3 osoby
6. Konsultanci – 3 osoby
7. Biuro rachunkowe

6.2 Harmonogram prac

Etapy mogą się składać z zadań.

Wskazać czasy trwania poszczególnych etapów i zadań – wykres Gantta.

obejmuje również harmonogram wdrożenia projektu – np. szkolenie, rozruch, konfiguracja, serwis – może obejmować różne wydania (tj. o różnej funkcjonalności – personal, professional, enterprise) i wersje (1.0, 1.5, itd.)



6.3 Etapy/kamienie milowe projektu

dla głównych etapów projektu

1. Sformułowanie planu wdrożenia systemu
2. Wybór technologii potrzebnej do zrealizowania tego projektu
3. Zdefiniowanie zasobów technicznych, które pozwalają nam na jego zrealizowanie
4. Stworzenie dokumentacji technicznej
5. Wdrożenie systemu

6. Konfiguracja i rozruch systemu
7. Testowanie systemu, co jest powiązane z jego konfiguracją
8. Podsumowanie zrealizowanych działań

7 Zarządzanie ryzykiem

7.1 Lista czynników ryzyka

1. Zignorowanie narzędzia przez filmowców
2. Brak zaufania poprzez opieranie się na społeczności
3. Narzędzia twórców kamer mogą być lepsze, bardziej rozbudowane, pomimo braku wsparcia innych kamer.
4. Brak dostrzeżenia narzędzia na rynku, w przypadku czego „poczta pantoflowa” się nie sprawdzi
5. Odchodzenie ludzi z projektu
6. Niezrozumienie potrzeb rynku

7.2 Ocena ryzyka

prawdopodobieństwo i wpływ

1. Ryzyko niezaistnienia narzędzia na rynku, niewykorzystania go przez filmowców i osób odpowiedzialnych za produkcję filmową. Wpłynie to niekorzystnie na twórców systemu pod względem finansowym i wizerunkowym.
2. Prawdopodobieństwo złego odbioru ze strony społeczności - ma to wpływ na zaufanie ze strony marki i wizerunek producentów.
3. Potencjalne ryzyko niezaistnienia aplikacji na rynku wśród innych dostępnych rozwiązań oferowanych przez twórców kamer, oznacza mniejszą funkcjonalność systemu względem rozwiązań dostępnych na rynku, wpłynie to na odbiór produktu i jego mniejszą popularność.
4. Niewielkie prawdopodobieństwo wystąpienia, ma wpływ na rozpromowanie produktu.
5. Potencjalne ryzyko związane z opuszczaniem zespołu przez członków. Ma to bezpośredni wpływ na przyszłość rozwoju firmy, jej finanse oraz wizerunek.
6. Małe prawdopodobieństwo na niezrozumienie potrzeb rynku, które wpłynęłoby na rozwój firmy, jakość wytwarzanego produktu oraz zaufanie klientów.

7.3 Plan reakcji na ryzyko

1. Zignorowanie narzędzia przez filmowców:
 - a. Próba skontaktowania się z ważnymi youtuberami, którzy mają spore powiązanie z kinematografią o ocenienie narzędzia, oraz próba rozwiązania problemu w sposób niesponsorowany z youtuberami – zaburzy to wizerunek narzędzia, oraz naszą wiarygodność
 - b. Oczekiwanie na skutki poczty pantoflowej – wiele przypadków było w historii, które na tym polegały np. pierwsze pokemony, czy telefony OnePlus. Jeżeli narzędzie będzie dobre to znajdą się fani marki, którzy będą polecać narzędzie.

- c. Stworzenie reklam pozwalających na rozpromowanie produktu – nawet wśród jedynie grona entuzjastów.
- 2. Brak zaufania poprzez opieranie się na społeczności:
 - a. Ryzyko jest jedynie możliwe, jeżeli zaniechamy pracę nad programem oraz jego rozwój na rzecz rozwiązań społeczności. Ogólnie relacja twórcy-społeczność, jeżeli wzajemnie siebie słuchają obie strony – może być korzystna.
 - b. Opieranie się na społeczności zazwyczaj jest korzystne i traktowane jako pozytywny aspekt, czyli można przez to rozumieć, że twórcy rozumieją swoich użytkowników.
- 3. Narzędzia twórców kamer mogą być lepsze, bardziej rozbudowane, pomimo braku wsparcia innych kamer:
 - a. Rozwiązywanie problemu z twórcami API, z „odpaleniem” im minimalnego procentu zysku, jeżeli użytkownik aplikacji korzysta z ich kamery, lub i bez niego.
 - b. Rozwiązanie problemu samodzielnie, analizując działanie tamtej aplikacji, oraz naniesienie tych rozwiązań do naszego projektu. Wraz z czasem nowe funkcjonalności będą znacznie lepiej działać oraz oferować dużo więcej.
- 4. Brak dostrzeżenia narzędzia na rynku, w przypadku czego „poczta pantoflowa” się nie sprawdzi:
 - a. Stworzenie reklam pozwalających na rozpromowanie produktu – nawet wśród jedynie grona entuzjastów.
- 5. Odchodzenie ludzi z projektu:
 - a. Próba zatrzymania ich za pomocą owocowych czwartków, oraz różnych benefitów takich jak karta multisport, praca w młodym i ambitnym zespole, oraz bardziej konkurencyjnej pensji w porównaniu do innych firm.
 - b. Jeżeli odejdą to próba znalezienia tak samo dobrze wykwalifikowanych pracowników, lub lepszych poprzez wykwalifikowaną, zatrudnioną do tego zewnętrzną kadrę HR, albo w przypadku kryzysu firmy – zatrudnienie tej osoby samemu.
- 6. Niezrozumienie potrzeb rynku:
 - a. Próba kontaktowania się ze społecznością, aby zanotować co powinniśmy zmienić w projekcie i nanieść te poprawki, lub zmienić kompletnie kierunek, w którym powinien zmierzać program

8 Zarządzanie jakością

8.1 Scenariusze i przypadki testowe

Numer	1		
Nazwa scenariusza	Wsparcie dla producentów wielu kamer		
Kategoria	Urządzenia		
Opis	sprawdzenie czy dany model kamery jest wspierany		
Tester	poszczególne użytkownik odpowiedniego modelu kamery		
Termin	przy każdej implementacji poszczególnej kamery w aplikacji		
Narzędzia wspomagające	Kamera cyfrowa		
Przebieg działań	L.p.	Działania testera	Działania systemu
	1	Tester uruchamia program	Programiści implementują API, oraz funkcjonalności konieczne do obsługi kamery
	2	Użytkownik podłącza kamerę do komputera	Program wykrywa sprzęt i w przypadku braku sterowników do tego urządzenia pobiera z serwera dane.
	3	Użytkownik korzysta z kamery	Aplikacja udostępnia narzędzia
	4	Komunikat „kamera niewspierana”, lub błędy podczas korzystania z aplikacji	Zgłoszenie niewspieranej kamery do administratora, który przekazuje informację dalej
	Założenia	Posiadanie kamery od danego producenta, która	

		wspiera tethering.		
	Środowisko	Oprogramowanie w wersji alpha.		
	Warunki wstępne	Kompatybilność pomiędzy poszczególnymi modelami kamer		
	Dane wejściowe	ustawienia konfiguracyjne urządzeń, poszczególnych kamer		
Zestaw danych testowych	Dane prawidłowe	Złe dane wejściowe	Opcja niedostępna	
	Zwraca: Pliki dziennika, a tester przechodzi do testowania reszty opcji	Zwraca: Pliki dziennika, oraz komunikat o niepoprawnym działaniu do testera	Zwraca: Pliki dziennika, informację od testera o posiadanej kamerze i działaniu	
Warunek zaliczenia testu	Wspieranie pełnej funkcjonalności kamery, zgodnie z wymaganiami aplikacji			

<i>Numer</i>		2		
<i>Nazwa scenariusza</i>		Podgląd na żywo z wielu kamer		
<i>Kategoria</i>		Urządzenia		
<i>Opis</i>		Przy podłączeniu kamer do stacji bazowej za pomocą kabla ethernetowego, jest możliwość podglądu na żywo z wielu kamer		
<i>Tester</i>		poszczególne użytkownik odpowiedniego modelu kamery, lub ich wielu podpiętych do stacji bazowej za pomocą kabla ethernetowego		
<i>Termin</i>		przy każdej implementacji poszczególnej kamery w aplikacji		
<i>Narzędzia wspomagające</i>		Kamera cyfrowa		
<i>Przebieg działań</i>		<i>L.p.</i>	<i>Działania testera</i>	<i>Działania systemu</i>
		1	Tester uruchamia program	Programiści implementują API, oraz funkcjonalności konieczne do obsługi kamery
		2	Użytkownik podłącza kamery do stacji bazowej	Program wykrywa sprzęt i w przypadku braku sterowników do tego urządzenia pobiera z serwera dane.
		3	Użytkownik ma możliwość korzystania z wielu kamer naraz i możliwości ich podglądu	Aplikacja udostępnia narzędzia
	<i>Założenia</i>	Posiadanie kamery od danego producenta, która wspiera tethering.		
	<i>Środowisko</i>	Oprogramowanie w wersji alpha.		
	<i>Warunki wstępne</i>	Kompatybilność pomiędzy poszczególnymi modelami kamer		
	<i>Dane wejściowe</i>	ustawienia konfiguracyjne urządzeń, poszczególnych kamer		
<i>Zestaw danych testowych</i>		<i>Dane</i>	<i>Złe dane</i>	<i>Opcja</i>

	<i>prawidłowe</i>	<i>wejściowe</i>	<i>niedostępna</i>
	<i>Zwraca: Pliki dziennika, a tester przechodzi do testowania reszty opcji</i>	<i>Zwraca: Pliki dziennika, oraz komunikat o niepoprawnym działaniu do testera</i>	<i>Zwraca: Pliki dziennika, informację od testera o posiadanej kamerze i działaniu</i>
<i>Warunek zaliczenia testu</i>	<i>Wspieranie pełnej funkcjonalności kamery, zgodnie z wymaganiami aplikacji</i>		

<i>Numer</i>	3		
<i>Nazwa scenariusza</i>	Testowanie funkcjonalności konfiguracji ustawień kamer za pomocą stacji bazowej		
<i>Kategoria</i>	Urządzenia		
<i>Opis</i>	Przy podłączeniu kamer do stacji bazowej za pomocą kabla ethernetowego, jest możliwość podglądu na żywo z wielu kamer		
<i>Tester</i>	poszczególne użytkownik odpowiedniego modelu kamery, lub ich wielu podpiętych do stacji bazowej za pomocą kabla ethernetowego		
<i>Termin</i>	przy każdej implementacji poszczególnej kamery w aplikacji		
<i>Narzędzia wspomagające</i>	kamery cyfrowe, stacja bazowa		
<i>Przebieg działań</i>	<i>L.p.</i>	<i>Działania testera</i>	<i>Działania systemu</i>
	1	Tester uruchamia program	Programiści implementują API, oraz funkcjonalności konieczne do obsługi kamery
	2	Użytkownik podłącza kamery do stacji bazowej	Program wykrywa sprzęt i w przypadku braku sterowników do tego urządzenia pobiera z serwera dane.
	3	Użytkownik ma możliwość korzystania z wielu kamer naraz i możliwości ich podglądu	Aplikacja udostępnia narzędzia
	4	Użytkownik zmienia parametry kamery z możliwością podglądu zmian na	Aplikacja udostępnia narzędzia

			bieżąco	
	<i>Założenia</i>	Posiadanie kamery od danego producenta, która wspiera tethering.		
	<i>Środowisko</i>	Oprogramowanie w wersji alpha.		
	<i>Warunki wstępne</i>	Kompatybilność pomiędzy poszczególnymi modelami kamer		
	<i>Dane wejściowe</i>	ustawienia konfiguracyjne urządzeń, poszczególnych kamer		
<i>Zestaw danych testowych</i>		<i>Dane prawidłowe</i>	<i>Złe dane wejściowe</i>	<i>Opcja niedostępna</i>
		Zwraca: Pliki dziennika, a tester przechodzi do testowania reszty opcji	Zwraca: Pliki dziennika, oraz komunikat o niepoprawnym działaniu do testera	Zwraca: Pliki dziennika, informację od testera o posiadanej kamerze i działaniu
<i>Warunek zaliczenia testu</i>		Poprawna konfiguracja urządzeń, Stabilność zastosowanych parametrów, synchronizacja ustawień		

Numer		4		
Nazwa scenariusza		Wspieranie urządzeń od renomowanych producentów akcesoriów		
Kategoria		Urządzenia		
Opis		Testowanie funkcjonalności dodawania akcesoriów do urządzeń, synchronizacji pomiędzy kamerą a akcesorium		
Tester		Operator kamery		
Termin		przy pierwszej implementacji akcesorium do kamery		
Narzędzia wspomagające		Kamera, akcesoria		
Przebieg działań		L.p.	Działania testera	Działania systemu
		1	Przygotowanie akcesorium do pierwszego sparowania z urządzeniem	Oczekiwanie na dodanie nowego modułu
		2	Podpięcie akcesorium do kamery	Zrealizowanie dodania nowego akcesosium, informacja o akcesorium w systemie
		3	Parowanie akcesorium z programem	Informacja o poprwniej konfiguracji akcesorium
	Założenia	Możliwość podłączenia akcesoriów wielu różnych producentów do kamer		
	Środowisko	Oprogramowanie w wersji alpha.		
	Warunki wstępne	Stan gotowości kamery na dodanie akcesorium		
	Dane wejściowe	Dane techniczne, konfiguracyjne związane z akcesorium		
Zestaw danych testowych		Dane prawidłowe	Złe dane wejściowe	
		Akcesorium umożliwiające sparowanie go z kamerą	Podpięcie akcesorium które nie jest wspierane z danym typem kamery	
Warunek zaliczenia testu		Poprawne zrealizowanie podłączenia akcesorium do kamery		

<i>Numer</i>	5		
<i>Nazwa scenariusza</i>	Możliwość zaprogramowania zachowania urządzeń (scenariuszy) we własnej wersji interfejsu „Scratchowego”.		
<i>Kategoria</i>	Urządzenia, skryptowanie		
<i>Opis</i>	Sprawdzenie czy interfejs skryptowania działa		
<i>Tester</i>	Zwykły użytkownik		
<i>Termin</i>	przy każdej implementacji nowości w interfejsie skryptowania		
<i>Narzędzia wspomagające</i>	Kamera, akcesoria		
<i>Przebieg działań</i>	<i>L.p.</i>	<i>Działania testera</i>	<i>Działania systemu</i>
	1	Tester przechodzi do interfejsu skryptowania	Wcześniej bardzo dokładnie skalibrowane urządzenie zostaje dodane do możliwości oskryptowania
	2	Tester tworzy skrypt	Interfejs scratchowy oferuje zaimplementowanie wielu urządzeń, zaprogramowanych ruchów w kilku osiach, obrotów etc.
	3	Po zaimplementowaniu skryptu tester wykorzystuje go przy kręceniu materiału w celach testu.	Informacja o poprawnie wykonywanych działaniach
	<i>Założenia</i>	Posiadanie kamery od danego producenta, która wspiera tethering.	
	<i>Środowisko</i>	Oprogramowanie w wersji alpha.	
	<i>Warunki wstępne</i>	Poprawnie skalibrowane urządzenia.	
	<i>Dane wejściowe</i>	Dane techniczne, konfiguracyjne związane z akcesorium	
<i>Zestaw danych testowych</i>	<i>Dane prawidłowe</i>	<i>Złe dane wejściowe</i>	<i>Opcja niedostępna</i>
	Zwraca: Pliki dziennika, a tester przechodzi do testowania	Zwraca: Pliki dziennika, oraz komunikat o niepoprawnym działaniu do testera	Zwraca: Pliki dziennika, informację od testera o posiadanej kamerze

	<i>reszty opcji</i>		<i>i działaniu</i>
<i>Warunek zaliczenia testu</i>	Wykonywanie poprawnie skryptu, zgodnie ze schematami		

<i>Numer</i>	6		
<i>Nazwa scenariusza</i>	Możliwość tworzenia własnych narzędzi typu follow Focus, jeźdźnik i sparowania go z programem.		
<i>Kategoria</i>	Urządzenia		
<i>Opis</i>	sprawdzenie czy API programu jest prawidłowe		
<i>Tester</i>	Zwykły użytkownik		
<i>Termin</i>	przy każdej implementacji nowości w API		
<i>Narzędzia wspomagające</i>	Kamera, akcesoria		
<i>Przebieg działań</i>	<i>L.p.</i>	<i>Działania testera</i>	<i>Działania systemu</i>
	1	Po podłączeniu urządzenia, należy wejść do menu urządzeń podłączonych do komputera, oraz go wybrać, oraz przypisać mu kategorię.	System odczytuje urządzenie.
	2	Po wybraniu kategorii należy skalibrować urządzenie, program będzie wykonywać „własne ruchy” i użytkownik, będzie wprowadzał co było efektem działania komputera.	System wykonuje własne ruchy i czeka na potwierdzenie ruchów przez testera.
	3	Po poprawnej kalibracji można zacząć już korzystać z własnego rozwiązania	Informacja o poprawnie wykonywanych działaniach
	<i>Założenia</i>	Posiadanie kamery od danego producenta, która wspiera tethering.	
	<i>Środowisko</i>	Oprogramowanie w wersji alpha.	
	<i>Warunki wstępne</i>	Zapoznanie się użytkownika z API, oraz możliwościami programu, stworzenie oraz podłączenie poprawnie działającego własnego akcesorium	
	<i>Dane wejściowe</i>	Dane techniczne, konfiguracyjne związane z akcesorium	
<i>Zestaw danych testowych</i>	<i>Dane prawidłowe</i>	<i>Złe dane wejściowe</i>	<i>Opcja niedostępna</i>
	Zwraca: Pliki	Zwraca: Pliki	Zwraca: Pliki

	<i>dziennika, a tester przechodzi do testowania reszty opcji</i>	<i>dziennika, oraz komunikat o niepoprawnym działaniu do testera</i>	<i>dziennika, informację od testera o posiadanej kamerze i działaniu</i>
<i>Warunek zaliczenia testu</i>	Poprawnie sparowane urządzenie i działające		

<i>Numer</i>		7		
<i>Nazwa scenariusza</i>		Testowanie funkcjonalności streamowania obrazu z kamery na żywo do telewizji, serwisów internetowych		
<i>Kategoria</i>		Streaming, transmisja zdalna		
<i>Opis</i>		Testowanie funkcjonalności zdalnej transmisji		
<i>Tester</i>		Administrator sieci komputerowych		
<i>Termin</i>		przy każdej implementacji poszczególnej kamery w aplikacji		
<i>Narzędzia wspomagające</i>		Kamera, Interfejs połączenia z serwerem streamingu		
<i>Przebieg działań</i>		<i>L.p.</i>	<i>Działania testera</i>	<i>Działania systemu</i>
		1	Połączenie kamery z serwerem streamingu	Komunikat o poprawności połączenia
		2	Konfiguracja ustawień streamowania: Jakość transmisji, Enkoder, Przepływność (bitrate), rozdzielczość obrazu	Zatwierdzenie ustawień i gotowość do zdalnej transmisji
		3	Uruchomienie zdalnej transmisji	Zrealizowanie poprawnej transmisji zgodnie z ustawieniami
		4	Zatrzymanie transmisji	Wyłączenie transmisji, przejście w stan gotowości do uruchomienia nowej transmisji
	<i>Założenia</i>	Posiadanie urządzeń umożliwiających zrealizowanie zdalnej transmisji		
	<i>Warunki wstępne</i>	Skonfigurowane urządzenia pod kątem transmisji zdalnej		

	<i>Dane wejściowe</i>	<i>Dane dotyczące ustawień kamery, ustawienia konfiguracyjne transmisji zdalnej</i>	
<i>Zestaw danych testowych</i>		<i>Dane prawidłowe</i>	<i>Złe dane wejściowe</i>
		<i>Poprawna konfiguracja ustawień kamery, ustawień dotyczących transmisji zdalnej</i>	<i>Niespójność w ustawieniach transmisji</i>
<i>Warunek zaliczenia testu</i>	Poprawne zrealizowanie konfiguracji ustawień kamery, ustawień transmisji zdalnej.		

Numer		8		
Nazwa scenariusza		Synchronizacja ustawień do chmury przy wykupieniu odpowiedniej subskrypcji przez klienta		
Kategoria		Przechowywanie danych na serwerze		
Opis		Testowanie funkcjonalności zapisu i synchronizacji danych w chmurze		
Tester		Administrator sieci komputerowych		
Termin		Przy każdej implementacji nowości w usługach synchronizacji.		
Narzędzia wspomagające		Kamera, inne urządzenia przechowujące dane konfiguracyjne. Interfejs połączenia z serwerem		
Przebieg działań		L.p.	Działania testera	Działania systemu
		1	Utworzenie przykładowego zestawu danych testowych dotyczących ustawień kamery	-
		2	Próba połączenia z serwerem danych	Komunikat o stanie połączenia urządzenia z serwerem
		3	Operacja synchronizacji ustawień, danych w chmurze	Zrealizowanie synchronizacji ustawień, danych z urządzeniem
		4	Pobranie danych z serwera	Komunikat o poprawności zrealizowania działania
	Założenia	Posiadanie urządzeń umożliwiających zrealizowanie zapisu danych do chmury, odczytu danych i synchronizacji		
	Warunki wstępne	Połączenie urządzenia z serwerem		
	Dane wejściowe	Ustawienia konfiguracyjne urządzeń		
Zestaw danych testowych		Dane prawidłowe	Złe dane wejściowe	
		Poprawna synchronizacja ustawień użytkownika.	Niespójność w danych przesłanych do chmury.	
Warunek zaliczenia testu		Poprawne zsynchronizowanie ustawień użytkownika.		

	ustawień transmisji zdalnej.
--	------------------------------

<i>Numer</i>	9		
<i>Nazwa scenariusza</i>	Możliwość nakładania LUTów na podglądany obraz z kamery		
<i>Kategoria</i>	Obraz, urządzenia		
<i>Opis</i>	Testowanie możliwości nakładania LUTów na podglądany obraz z kamery		
<i>Tester</i>	Zwykły użytkownik		
<i>Termin</i>	Przy implementacji poszczególnych LUTów.		
<i>Narzędzia wspomagające</i>	Kamera		
<i>Przebieg działań</i>	<i>Lp.</i>	<i>Działania testera</i>	<i>Działania systemu</i>
	1.	Użytkownik wybiera z listy zaimportowanych/wbudowanych LUTów, taki który chciałby nałożyć na obraz z kamery.	Wysświetla listę LUTów.
	2.	Aby wybrać LUT należy kliknąć prawym przyciskiem myszy LUT i wybrać opcję „nałóż”	Aplikacja nakłada LUT na podgląd na żywo
	<i>Założenia</i>	Posiadanie kamery podpiętej do komputera, uruchomiona aplikacja.	
	<i>Warunki wstępne</i>	Posiadanie wspieranej kamery	
	<i>Dane wejściowe</i>	LUTy, obraz z kamery	
<i>Zestaw danych testowych</i>	<i>Dane prawidłowe</i>	<i>Złe dane wejściowe</i>	<i>Opcja niedostępna</i>
	Zwraca: Pliki dziennika, a tester przechodzi do testowania reszty opcji	Zwraca: Pliki dziennika, oraz komunikat o niepoprawnym działaniu do testera	Zwraca: Pliki dziennika, informację od testera o posiadanej kamerze i działaniu
<i>Warunek zaliczenia testu</i>	Nakładanie LUTów na obraz działa.		

<i>Numer</i>		<i>10</i>		
<i>Nazwa scenariusza</i>		Narzędzie posiada listę zalecanych pluginów, oraz wbudowany sklep z nimi, wraz z opiniami innych użytkowników		
<i>Kategoria</i>		<i>Aplikacja, Sieć komputerowa</i>		
<i>Opis</i>		Sprawdzenie działania sklepu		
<i>Tester</i>		Zwykły użytkownik		
<i>Termin</i>		przy każdej zmianie w działaniu sklepu		
<i>Narzędzia wspomagające</i>		<i>brak</i>		
<i>Przebieg działań</i>		<i>L.p.</i>	<i>Działania testera</i>	<i>Działania systemu</i>
		<i>1</i>	Użytkownik otwiera wbudowany sklep, otwiera listę zalecanych pluginów.	Program pobiera dane z internetu i wyświetla je wewnątrz aplikacji.
		<i>2</i>	Użytkownik pobiera plugin i go aktywuje w głównej części programu.	Program pobiera z serwera plugin.
		<i>3</i>	Jeżeli plugin będzie dobry, to użytkownik może pochwalić autorów, a jeżeli zły to może zostawić konstruktywną, jak i mniej krytykę.	Program przesyła opinię do sklepu.
	<i>Założenia</i>	Połączenie z internetem, działająca aplikacja.		
	<i>Środowisko</i>	Oprogramowanie w wersji alpha.		
	<i>Warunki wstępne</i>	Połączenie z internetem, zalogowane konto użytkownika, działająca aplikacja.		
	<i>Dane wejściowe</i>	Ustawienia aplikacji.		
<i>Zestaw danych testowych</i>		<i>Dane prawidłowe</i>	<i>Złe dane wejściowe</i>	<i>Opcja niedostępna</i>
		Zwraca: Pliki dziennika, a tester	Zwraca: Pliki dziennika, oraz komunikat o	Zwraca: Pliki dziennika, informacje od

	<i>przechodzi do testowania reszty opcji</i>	<i>niepoprawnym działaniu do testera</i>	<i>testera o posiadanej kamerze i działaniu</i>
<i>Warunek zaliczenia testu</i>	<i>Poprawnie działający plugin, sklep, oraz możliwość zostawienia opinii w sklepie.</i>		

<i>Numer</i>		<i>11</i>		
<i>Nazwa scenariusza</i>		Możliwość importu pluginów z jakichkolwiek innych źródeł za pomocą pliku, jak i jego eksportu z programu.		
<i>Kategoria</i>		<i>Aplikacja. Usprawnienia</i>		
<i>Opis</i>		Sprawdzenie czy zaimportowane pluginy/eksportowane działają		
<i>Tester</i>		Zwykły użytkownik		
<i>Termin</i>		przy każdej implementacji nowej wersji oprogramowania		
<i>Narzędzia wspomagające</i>		<i>brak</i>		
<i>Przebieg działań</i>		<i>L.p.</i>	<i>Działania testera</i>	<i>Działania systemu</i>
		1	Użytkownik otwiera listę pluginów, wybiera opcje importu pluginów	Program wyświetla dane z komputera wewnątrz aplikacji.
		2	Użytkownik importuje plugin i go aktywuje w głównej części programu.	Program importuje plugin
		3	Jeżeli plugin będzie dobry, to użytkownik może podzielić się nim eksportując go (jeżeli nie był płatny w sklepie).	Program umożliwia eksportowanie pluginów, aby dzielić się z innymi.
	<i>Założenia</i>	Działająca aplikacja		
	<i>Środowisko</i>	Oprogramowanie w wersji alpha.		
	<i>Warunki wstępne</i>	działająca aplikacja.		
	<i>Dane wejściowe</i>	Ustawienia aplikacji.		
<i>Zestaw danych testowych</i>		<i>Dane prawidłowe</i>	<i>Złe dane wejściowe</i>	<i>Opcja niedostępna</i>
		Zwraca: Pliki dziennika, a tester przechodzi do testowania	Zwraca: Pliki dziennika, oraz komunikat o niepoprawnym działaniu do	Zwraca: Pliki dziennika, informację od testera o posiadanej

	<i>reszty opcji</i>	<i>testera</i>	<i>kamerze i działaniu</i>
<i>Warunek zaliczenia testu</i>	<i>Poprawnie działający plugin,import, oraz eksport pluginu.</i>		

<i>Numer</i>		<i>12</i>		
<i>Nazwa scenariusza</i>		Możliwość przesłania własnego pluginu do weryfikacji przez twórców/administrację pod kątem awaryjności/działania/braku zagrożeń i możliwość wrzucenia tego do sklepu, oraz czerpania z tego korzyści (70% dla twórców pluginu, 30% dla twórców aplikacji)		
<i>Kategoria</i>		<i>Aplikacja, Sieć komputerowa</i>		
<i>Opis</i>		Sprawdzenie działania sklepu		
<i>Tester</i>		Administrator		
<i>Termin</i>		przy każdej implementacji nowości w sklepie		
<i>Narzędzia wspomagające</i>		<i>brak</i>		
<i>Przebieg działań</i>		<i>L.p.</i>	<i>Działania testera</i>	<i>Działania systemu</i>
		1	Użytkownik przesyła swój plugin do weryfikacji, ustala kwotę jaką chce otrzymywać za swoją pracę (za darmo/płatne)	System przesyła plugin do weryfikacji przez administratorów w sandboxie, oraz odbierają dane z formularza.
		2	Administratorzy weryfikują w piaskownicy plugin, jeżeli nie ma żadnych zagrożeń to przekazywany jest do ostatniego etapu.	Po weryfikacji pluginu system przetwarza dane z formularza i przesyła plugin do sklepu.
		3	Plugin pojawia się w sklepie, inni użytkownicy mogą go pobierać, a twórca może czerpać z tego korzyści.	<i>Serwer przetwarza żądania pobrania aplikacji.</i>
	<i>Założenia</i>	Połączenie z internetem, działająca aplikacja.		
	<i>Środowisko</i>	Oprogramowanie w wersji alpha.		
	<i>Warunki wstępne</i>	Połączenie z internetem, zalogowane konto użytkownika, działająca aplikacja.		

	<i>Dane wejściowe</i>	<i>Ustawienia aplikacji.</i>		
<i>Zestaw danych testowych</i>	<i>Dane prawidłowe</i>	<i>Złe dane wejściowe</i>	<i>Opcja niedostępna</i>	
	<i>Zwraca: Pliki dziennika, a tester przechodzi do testowania reszty opcji</i>	<i>Zwraca: Pliki dziennika, oraz komunikat o niepoprawnym działaniu do testera</i>	<i>Zwraca: Pliki dziennika, informację od testera o posiadanej kamerze i działaniu</i>	
<i>Warunek zaliczenia testu</i>	<i>Poprawnie działające przesyłanie pluginu do sklepu, oraz formularza z danymi.</i>			

<i>Numer</i>	<i>13</i>		
<i>Nazwa scenariusza</i>	Możliwość nakładania na nagrywany materiał informacji potrzebnych dla montażysty (scena, ujęcie, etc.)		
<i>Kategoria</i>	Aplikacja		
<i>Opis</i>	Sprawdzenie czy program może nanieść		
<i>Tester</i>	Zwykły użytkownik		
<i>Termin</i>	przy pierwszej implementacji tego rozwiązania w aplikacji.		
<i>Narzędzia wspomagające</i>	<i>brak</i>		
<i>Przebieg działań</i>	<i>L.p.</i>	<i>Działania testera</i>	<i>Działania systemu</i>
	<i>1</i>	Użytkownik wybiera opcje EXIF/DATA przy widoku wybranego materiału.	Aplikacja pokazuje informacje o materiale wideo, oraz umożliwia jego edycję
	<i>2</i>	Użytkownik może nanieść informację na klip potrzebnych na rzecz post produkcji.	Program nadpisuje/zapisuje informacje, które będą potrzebne w postprodukcji.
	<i>Założenia</i>	dostęp do nagranego/nagrywanego materiału, uruchomiona aplikacja.	
	<i>Środowisko</i>	Oprogramowanie w wersji alpha.	
	<i>Warunki wstępne</i>	dostęp do nagranego/nagrywanego materiału, uruchomiona aplikacja.	
	<i>Dane wejściowe</i>	<i>Dane, które chce się wprowadzić do materiału, materiał wideo.</i>	
<i>Zestaw danych testowych</i>	<i>Dane prawidłowe</i>	<i>Złe dane wejściowe</i>	<i>Opcja niedostępna</i>
	Zwraca: Pliki dziennika, a tester przechodzi do testowania reszty opcji	Zwraca: Pliki dziennika, oraz komunikat o niepoprawnym działaniu do testera	Zwraca: Pliki dziennika, informację od testera o posiadanej kamerze i działaniu
<i>Warunek zaliczenia testu</i>	klip ma naniesione informacje, które są potrzebne		

	montażysty i innym
--	--------------------

<i>Numer</i>		<i>14</i>		
<i>Nazwa scenariusza</i>		Wbudowane narzędzia do kalkulacji crop factor matrycy przy wybraniu odpowiedniej rozdzielczości/proporcji obrazu, kalkulacji czasu migawki/shutter angle (aby uzyskać obraz bez migotania w danej części świata (60Hz w PA, 50Hz w Europie i przynajmniej większej części Azji), kalkulacji ekspozycji (parametry wejściowe: shutter angle, filtr nd, frame rate, transmisja światła obiektywu)		
<i>Kategoria</i>		<i>Aplikacja, narzędzia</i>		
<i>Opis</i>		Sprawdzenie czy narzędzie do przeliczeń działa		
<i>Tester</i>		Zwykły użytkownik		
<i>Termin</i>		przy pierwszej implementacji tego rozwiązania w aplikacji.		
<i>Narzędzia wspomagające</i>		<i>brak</i>		
<i>Przebieg działań</i>		<i>L.p.</i>	<i>Działania testera</i>	<i>Działania systemu</i>
		<i>1</i>	Użytkownik wybiera, które działanie chce wykonać np. przeliczyć crop factor matrycy przy wybraniu innych proporcji ekranu. Aplikacja sama przelicza jaką rozdzielczość otrzymamy, crop factor, oraz inne informacje.	Program przetwarza obliczenia i zwraca je użytkownikowi.
	<i>Założenia</i>	uruchomiona aplikacja		
	<i>Środowisko</i>	Oprogramowanie w wersji alpha.		
	<i>Warunki wstępne</i>	uruchomiona aplikacja		
	<i>Dane wejściowe</i>	<i>Dane wprowadzone przez użytkownika</i>		
<i>Zestaw danych testowych</i>		<i>Dane prawidłowe</i>	<i>Złe dane wejściowe</i>	<i>Opcja niedostępna</i>
		Zwraca: Pliki dziennika, a tester przechodzi do testowania	Zwraca: Pliki dziennika, oraz komunikat o niepoprawnym działaniu do	Zwraca: Pliki dziennika, informację od testera o posiadanej

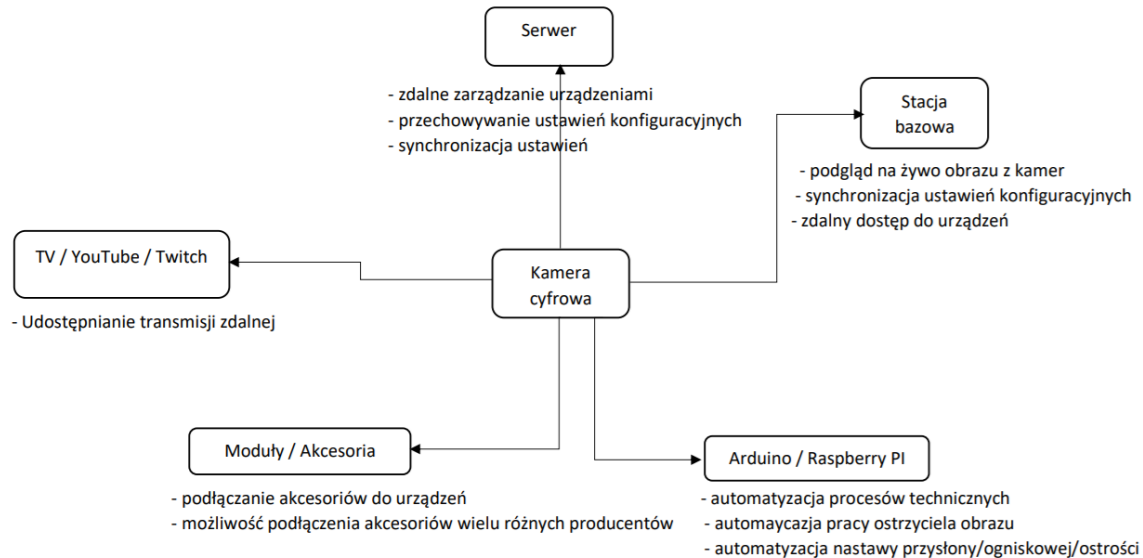
	<i>reszty opcji</i>	<i>testera</i>	<i>kamerze i działaniu</i>
<i>Warunek zaliczenia testu</i>	<i>Wszystkie wyniki są poprawne z danych działań.</i>		

9 Projekt techniczny

9.1 Opis architektury systemu

z ew. rysunkami pomocniczymi

- System jest kompatybilny z kamerami cyfrowymi oraz analogowymi (w ograniczonym zakresie)
- System umożliwia automatyzację procesów technicznych zachodzących na planie:
 - automatyzacja pracy ostrzyciela obrazu
 - automatyzacja pracy osób odpowiedzialnych za nastawy przysłony/ogniskowej/ostrości
 - moduł ten jest zrealizowany za pośrednictwem technologii Arduino lub raspberry PI.
- System umożliwia podgląd na żywo obrazu z kamer:
 - przy podłączeniu kamery do stacji bazowej za pomocą kabla Ethernet
 - możliwość podglądu obrazu z wielu kamer jednocześnie
- Możliwość konfiguracji ustawień kamer za pomocą stacji bazowej
- System umożliwia udostępnianie transmisji zdalnej do serwisów internetowych (Youtube, Twitch) oraz telewizji
- Możliwość zdalnego dostępu do systemu przez Internet:
 - synchronizowanie ustawień w chmurze
 - zdalny dostęp do urządzeń
 - przechowywanie ustawień konfiguracyjnych na serwerze
 - zdalne zarządzanie urządzeniami
- System umożliwia dołączanie modułów/akcesoriów od renomowanych producentów urządzeń:
 - podłączenie akcesoriów do urządzeń
 - możliwość podłączenia akcesoriów wielu różnych producentów



9.2 Technologie implementacji systemu

W założeniach programu zakładamy, że program dla użytkownika będzie pisany w językach:

C++ dla kompilacji Windowsowej, w przypadku macOS – Objective-C, częściowo Swift. Swift umożliwia możliwie pisanie w kodzie najwyższego poziomu, przy zachowaniu tej samej wydajności co Objective-C, oraz posiadana biblioteka do tworzenia interfejsu użytkownika taki jak SwiftUI. W celu szybszej implementacji programu będzie używany język Python wraz z bibliotekami PyQt5, którego szybsze zaimplementowanie będzie miało skutki uboczne w wydajności aplikacji.

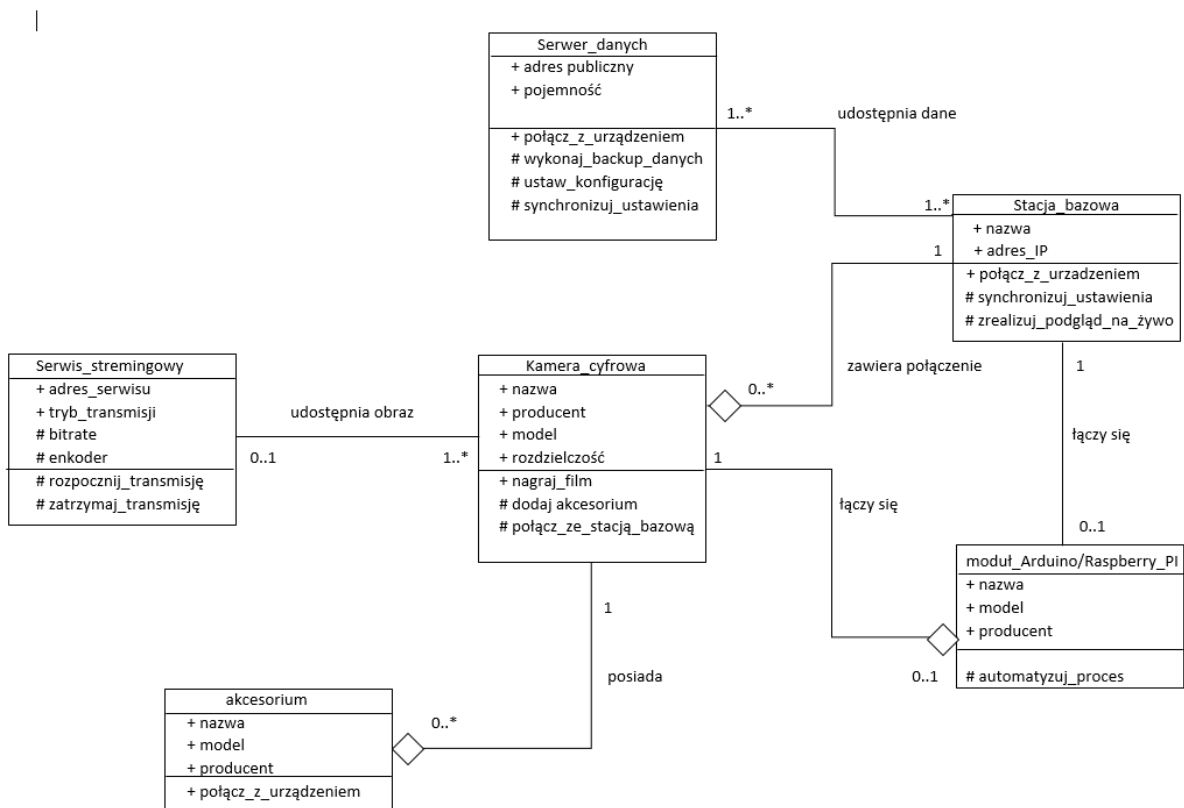
Natomiast dla serwera będzie program pisany w PHPie, z bazą danych w SQL. PHP został wybrany, ponieważ umożliwia on współpracę z SQLem, oraz z serwisami WWW. Również do zarządzania systemem baz danych został wybrany SQL, ponieważ jest efektywny przy pobieraniu danych.

Technologie te, zostały wybrane ze względu na ich wydajność, bezpieczeństwo, oraz przez to, że zespół ma znajomość tych języków programowania/języków zapytań.

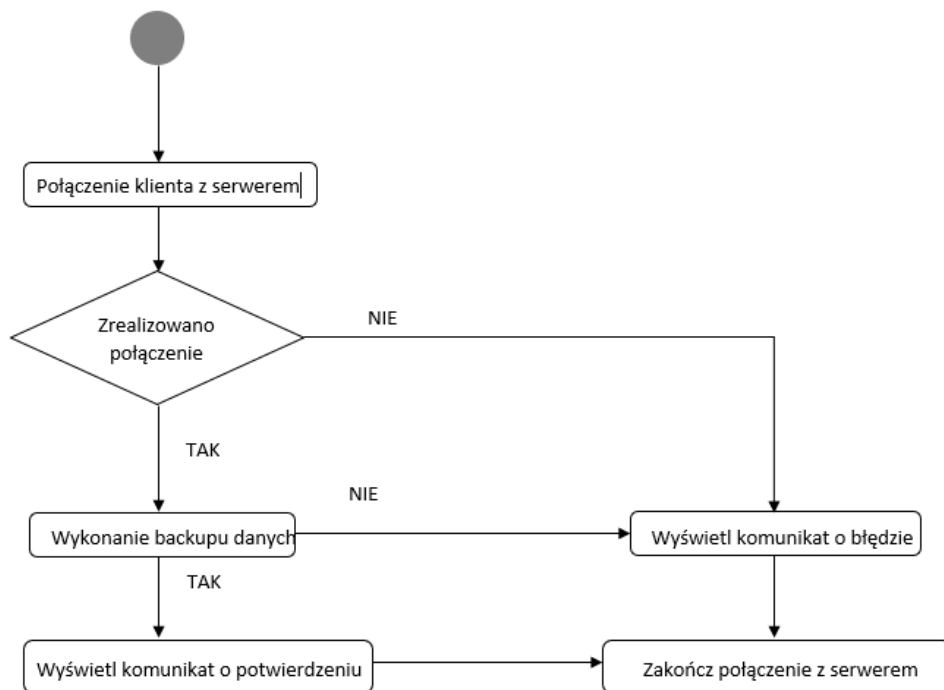
9.3 Diagramy UML

każdy diagram ma mieć tytuł oraz ma być na osobnej stronie diagramy przypadków użycia umieszczone w punkcie 5.2.2, a nie tutaj.

9.3.1 Diagram(-y) klas

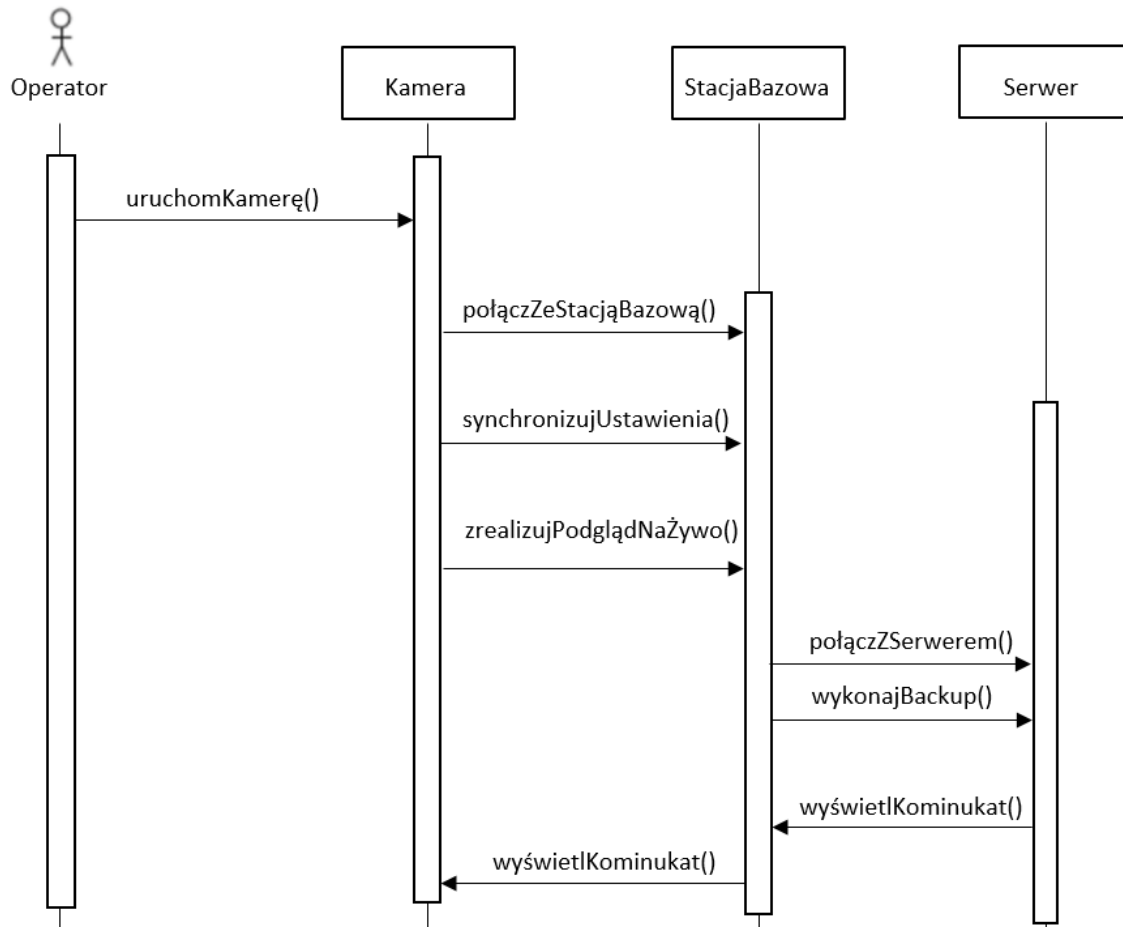


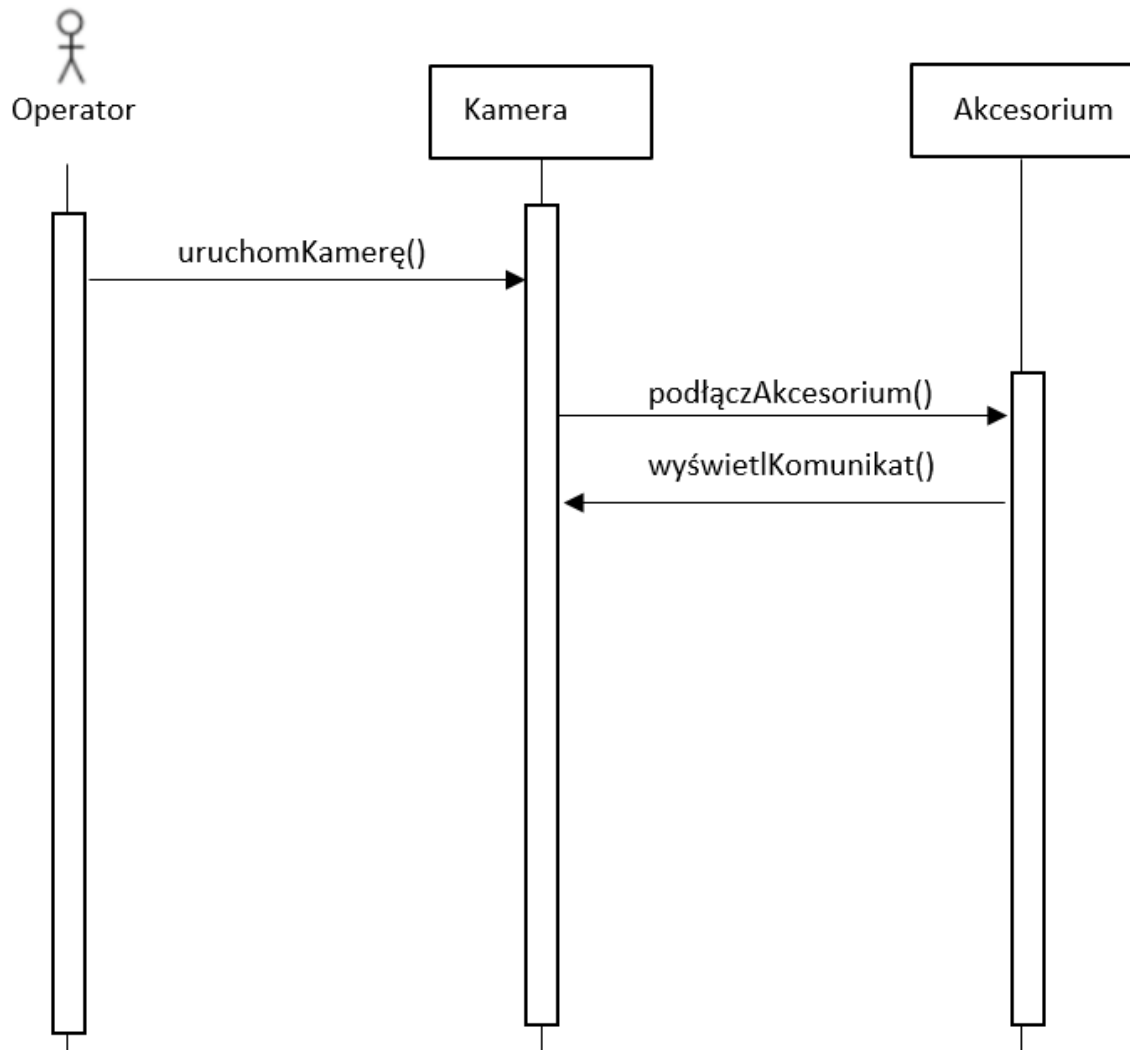
9.3.2 Diagram(-y) czynności

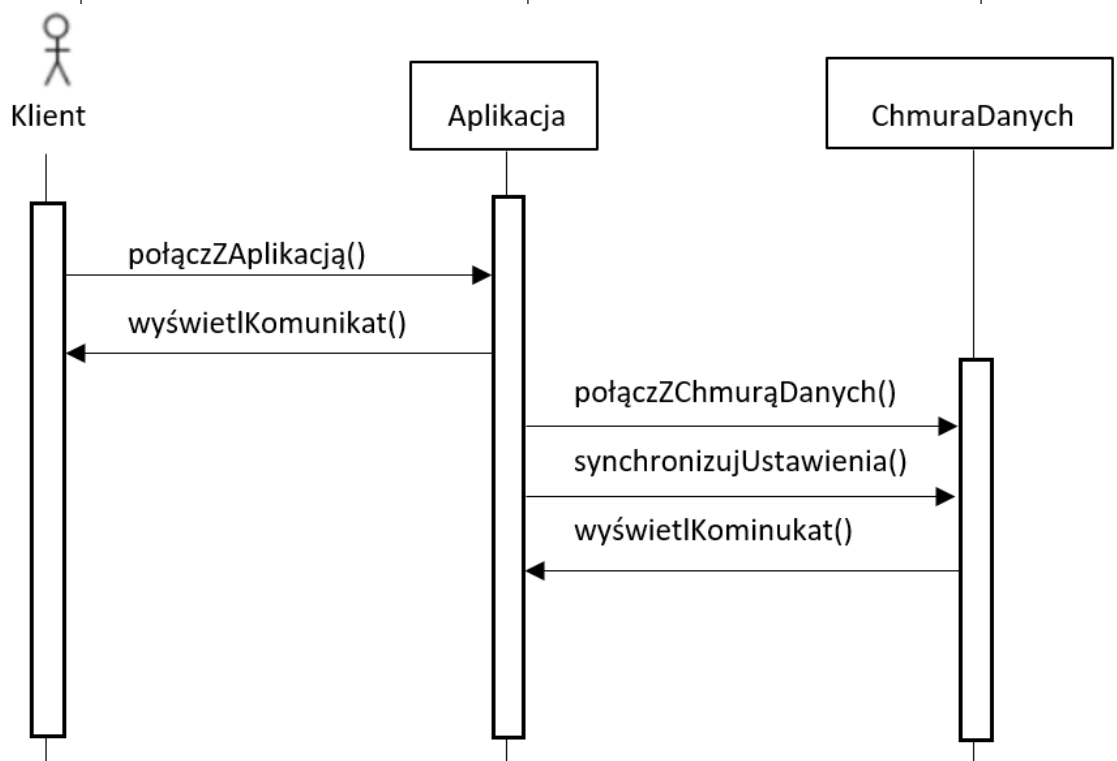
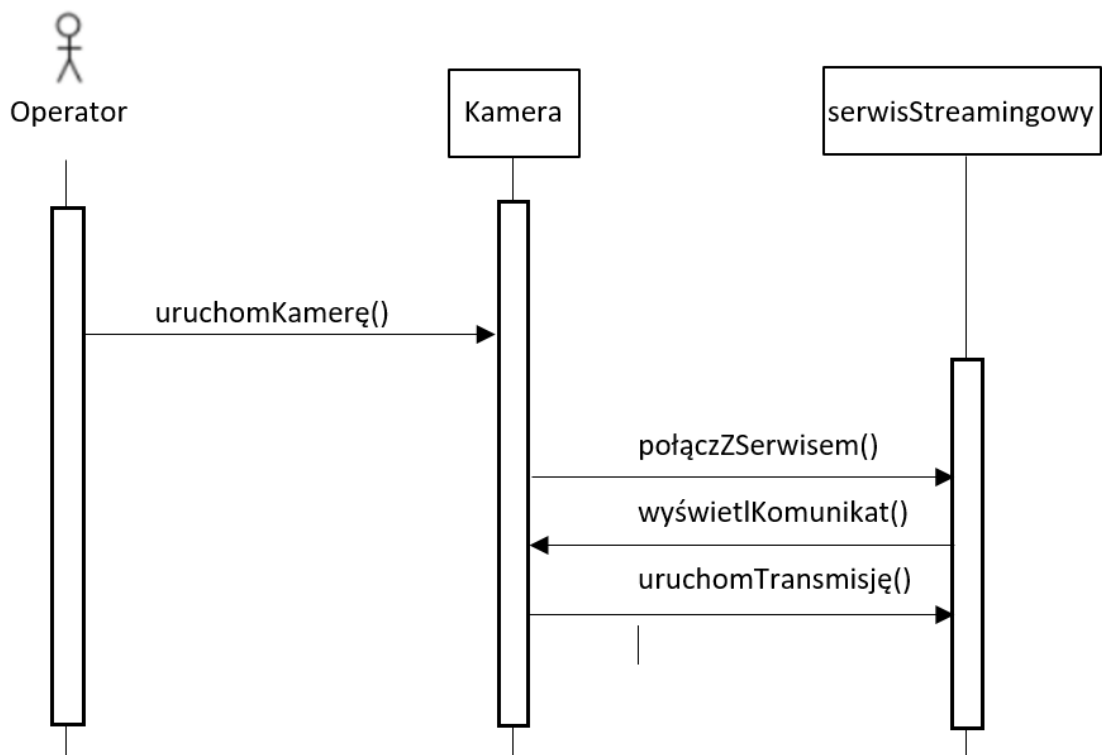


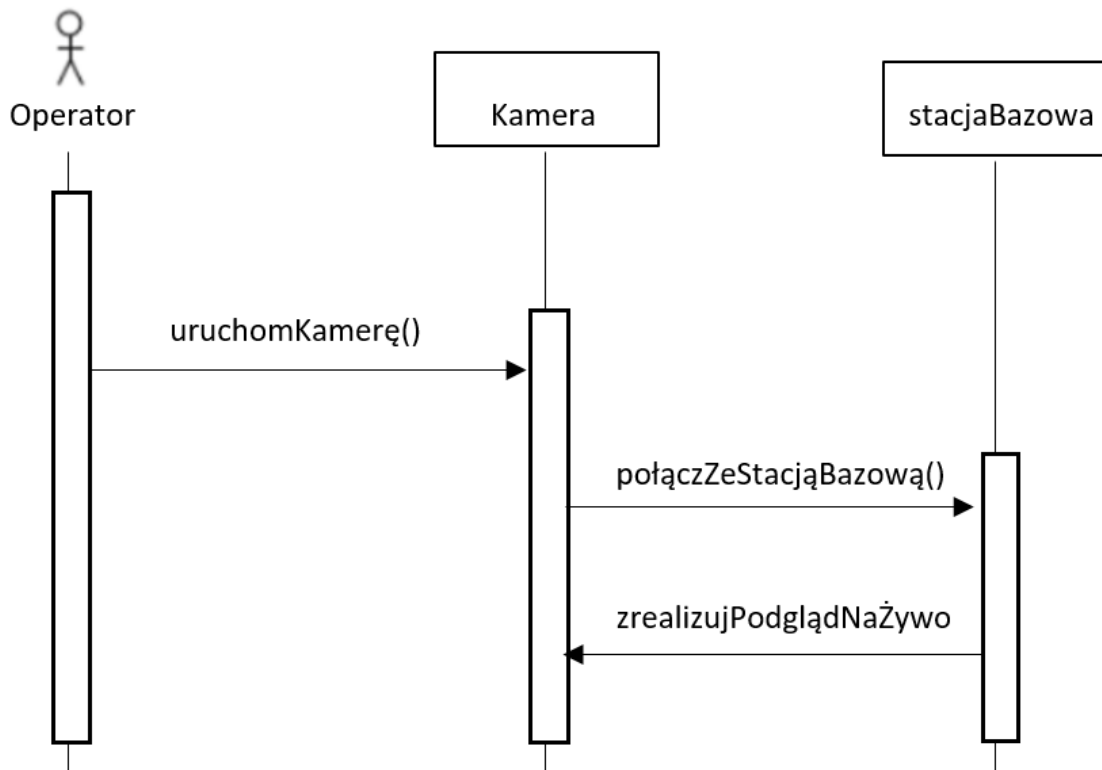
9.3.3 Diagramy sekwencji

co najmniej 5, w tym co najmniej 1 przypadek użycia zilustrowany kilkoma diagramami sekwencji



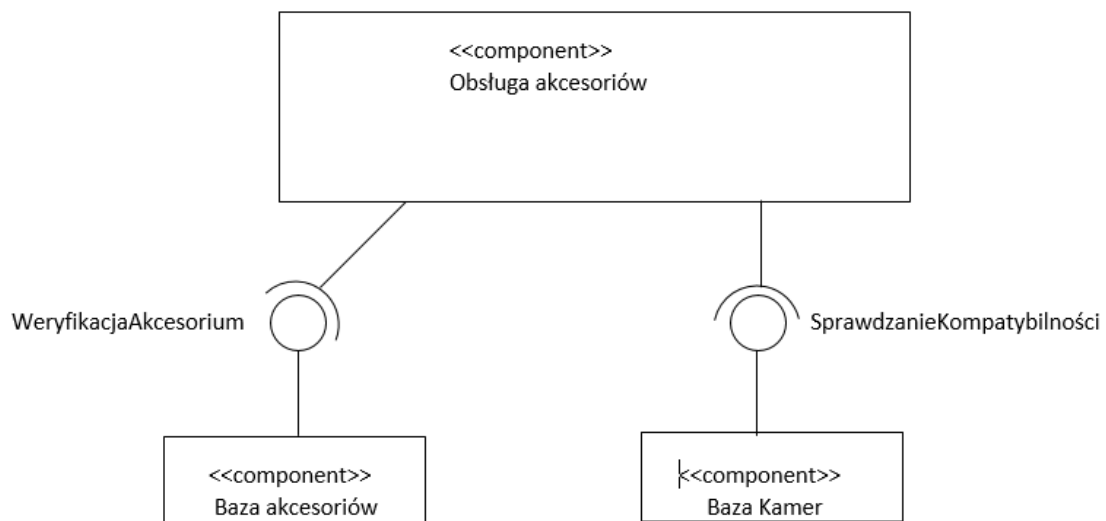






9.3.4 Inne diagramy

co najmniej trzy – komponentów, rozmieszczenia, maszyny stanowej itp.



9.4 Charakterystyka zastosowanych wzorców projektowych

informacja opisowa wspomagana diagramami (odsyłaczami do diagramów UML); jeśli wykorzystano wzorce projektowe, to należy wykazać dwa z nich

9.5 Projekt bazy danych

9.5.1 Schemat

w trzeciej formie normalnej; jeśli w innej to umieć uzasadnić wybór

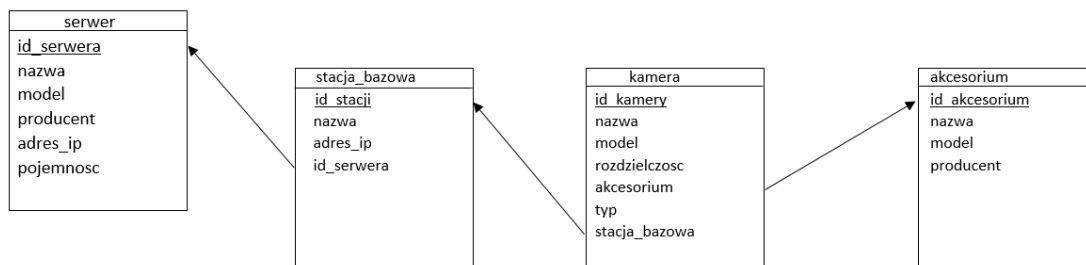
serwer(id_serwera, nazwa, model, producent, adres_ip, pojemnosc)

stacja_bazowa(id_stacji, nazwa, adres_ip, id_serwera)

kamera(id_kamery, nazwa, model, rozdzielczosc, akcesorium, typ, stacja_bazowa)

akcesorium(id_akcesorium, nazwa, model, producent)

9.5.2 Projekty szczegółowe tabel



9.6 Projekt interfejsu użytkownika

9.6.1 Lista głównych elementów interfejsu

1. Okno główne
2. Logowanie
3. Rejestracja
4. Ustawienia
5. Okno podglądu
6. Okno sklepu
7. Okno LUTów
8. Interfejs skryptowania
9. Okno przeliczeń

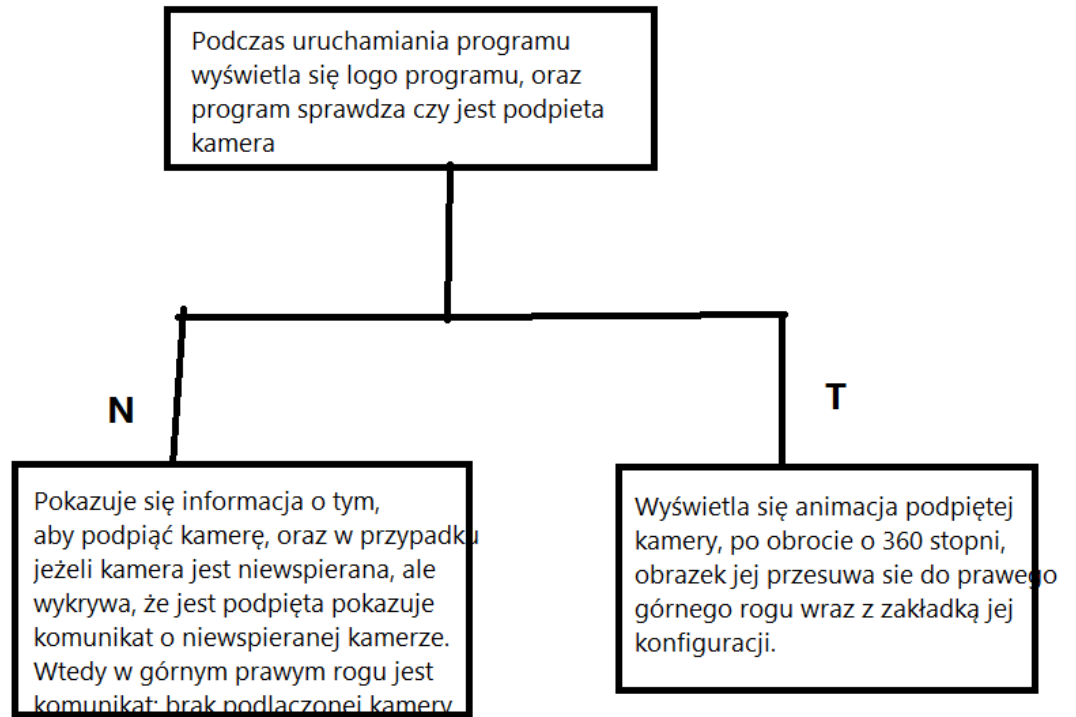
9.6.2 Przejścia między głównymi elementami

Otworzenie aplikacji -> Logowanie/Rejestracja -> Okno główne

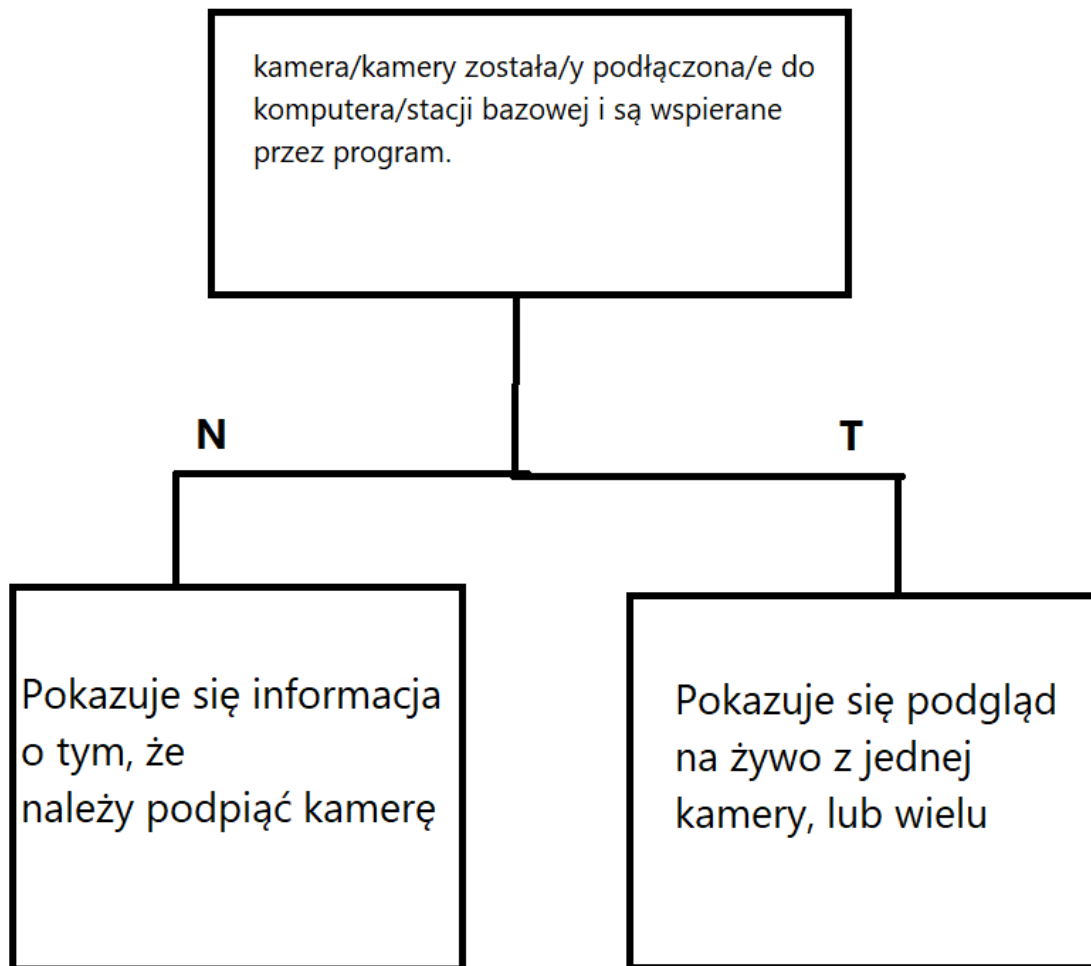
Okno główne -> Ustawienia; Okno podglądu; Okno sklepu; Okno LUTów; Interfejs Skryptowania; Okno przeliczeń

9.6.3 Projekty szczegółowe poszczególnych elementów

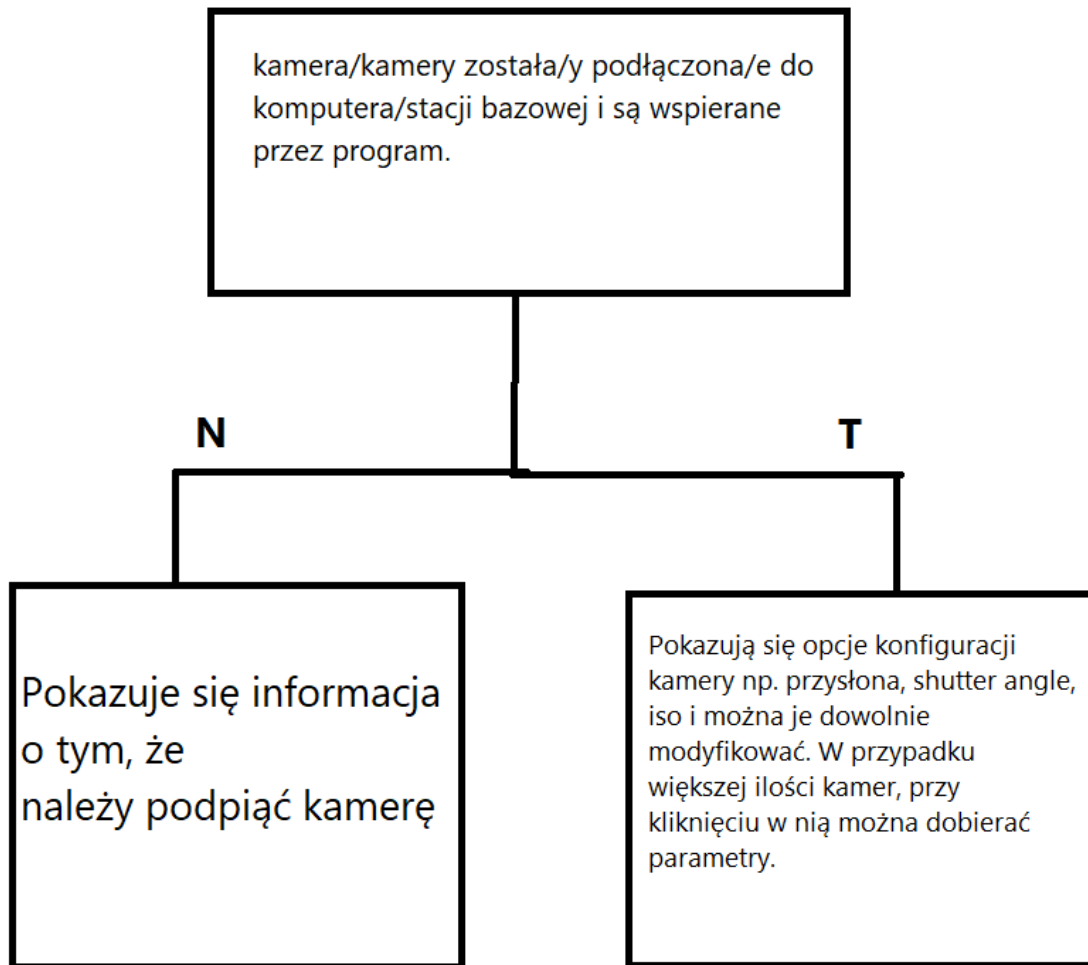
- numer – 1
- nazwa – Wsparcie dla producentów wielu kamer
- projekt graficzny:



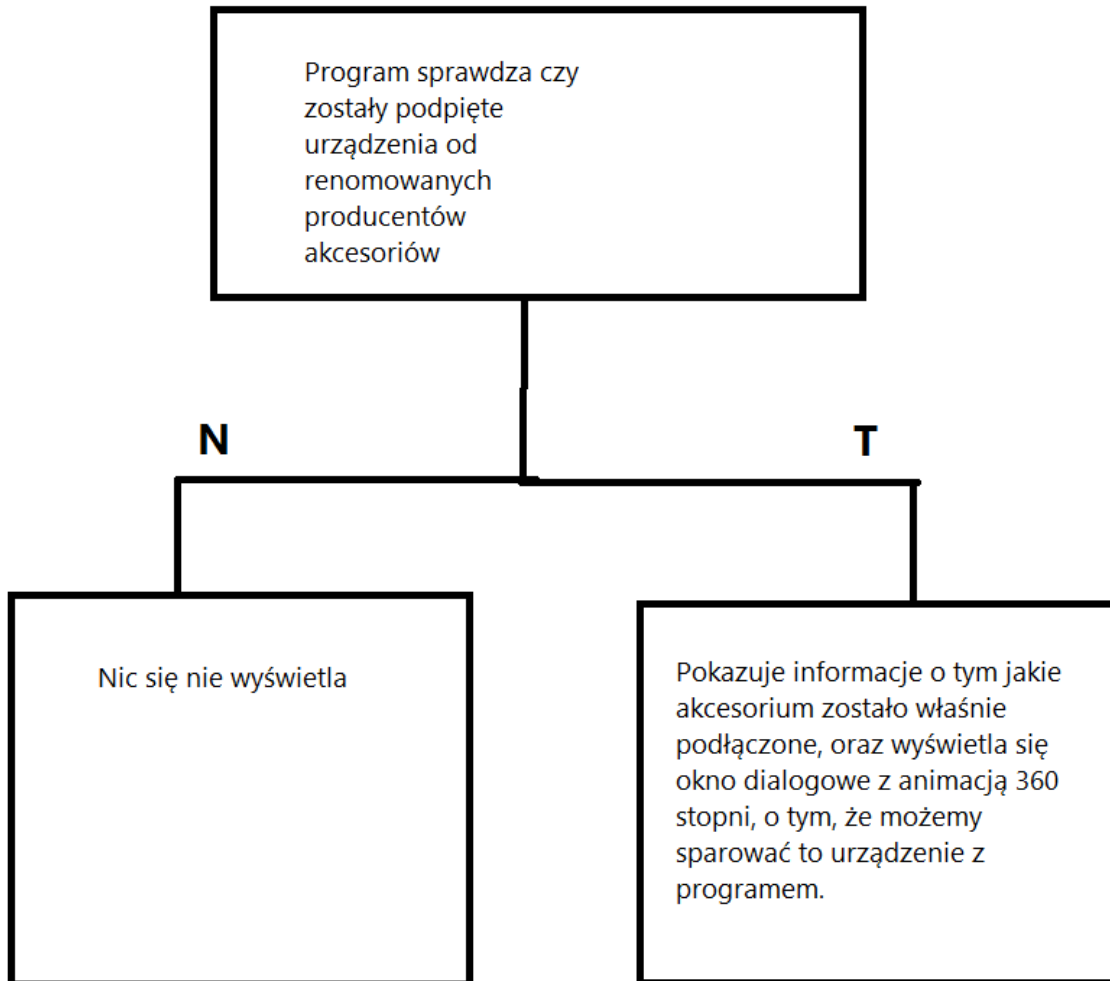
- numer – 2
- nazwa – Przy podłączeniu kamer do stacji bazowej za pomocą kabla ethernetowego, jest możliwość podglądu na żywo z wielu kamer
- projekt graficzny:



- numer – 3
- nazwa – Możliwość konfiguracji ustawień kamery, lub wielu kamer (za pomocą stacji bazowej)
- projekt graficzny:



- numer – 4
- nazwa – Wspieranie urządzeń od renomowanych producentów akcesoriów
- projekt graficzny:



- numer – 5
- nazwa – Możliwość zaprogramowania zachowania urządzeń (scenariuszy) we własnej wersji interfejsu „Scratchowego”.
- projekt graficzny:

Wchodzisz do interfejsu skryptowania



Wybierasz odpowiedni blok dla schematu
(ify, deklaracje zmiennych etc.)

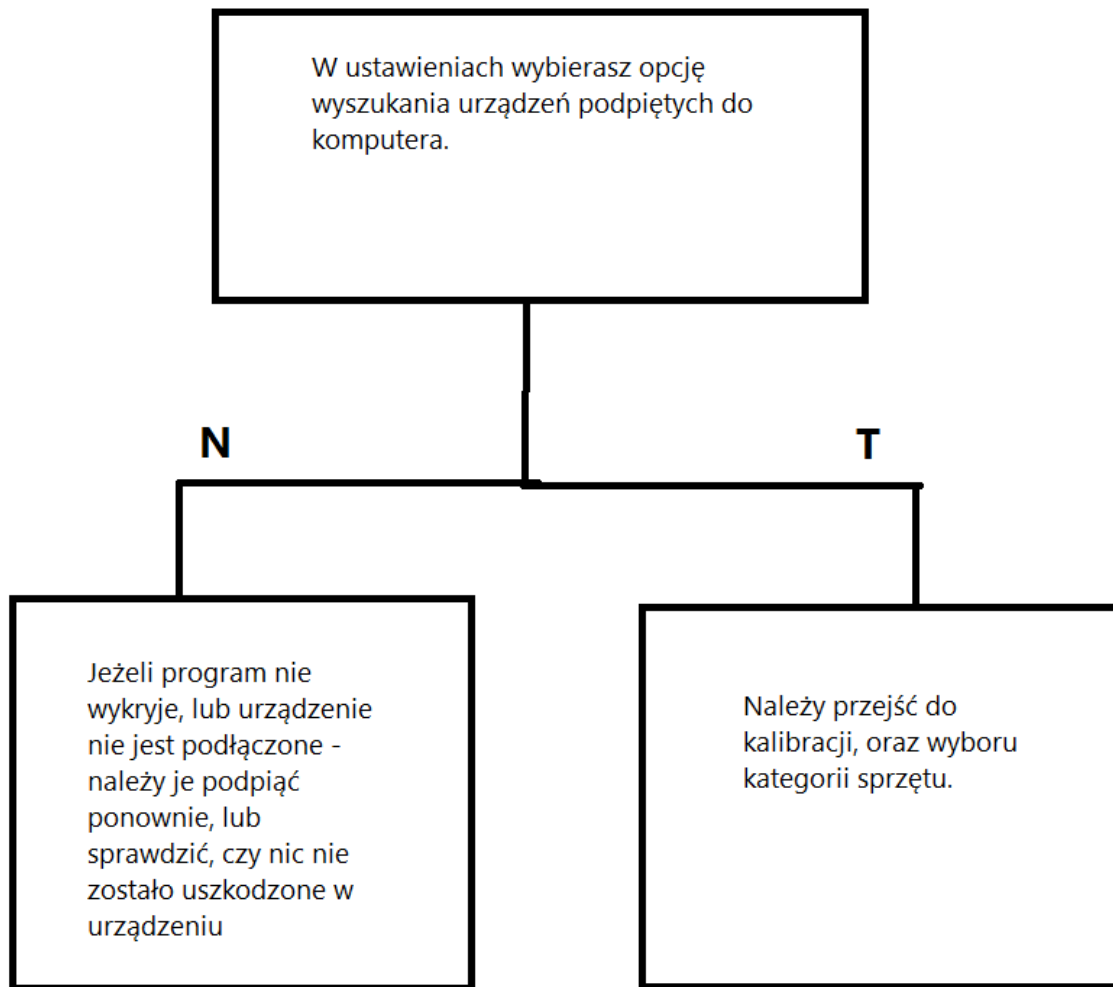


Konstruujesz algorytm

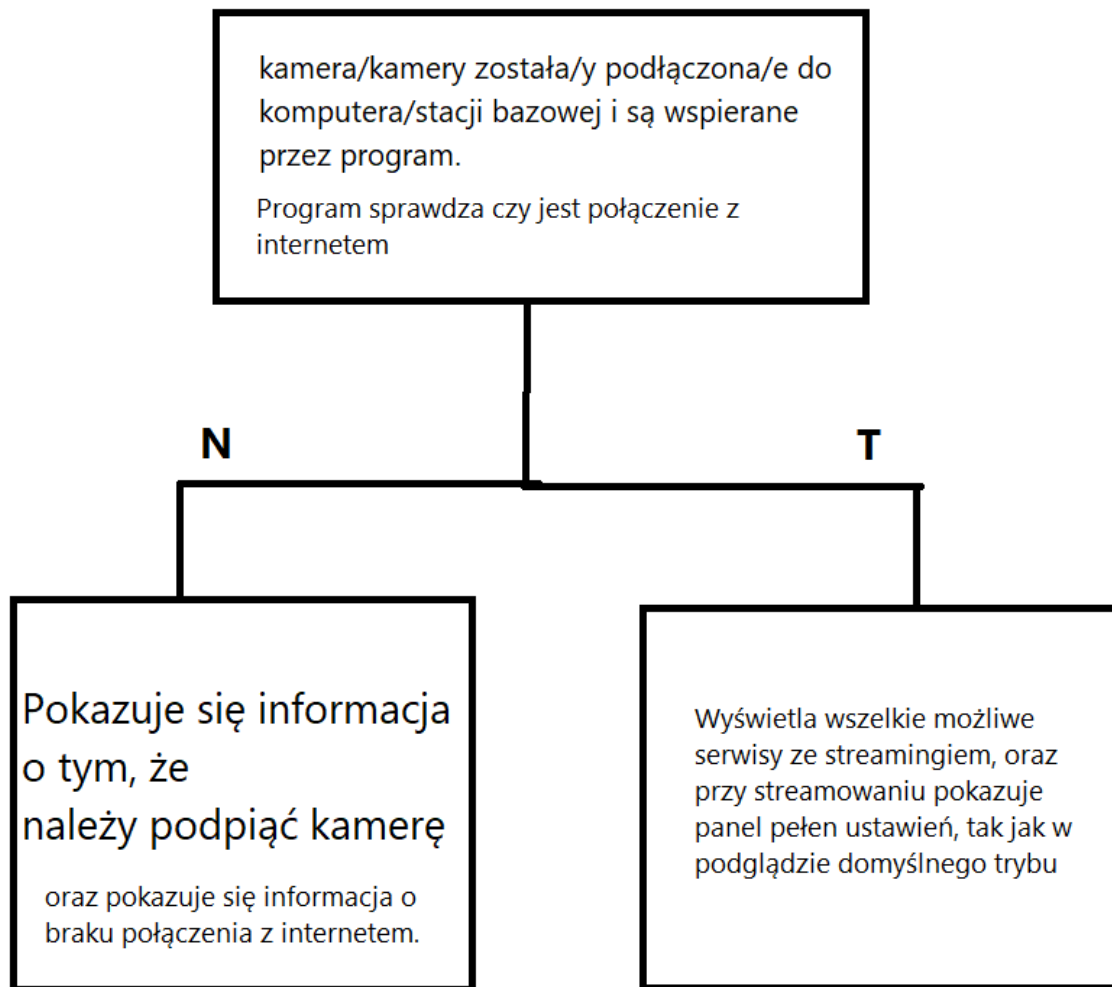


Po skonstruowaniu algorytmu/skryptu
zatwierdzasz go i zaczynasz z niego
korzystać przy kręceniu filmów

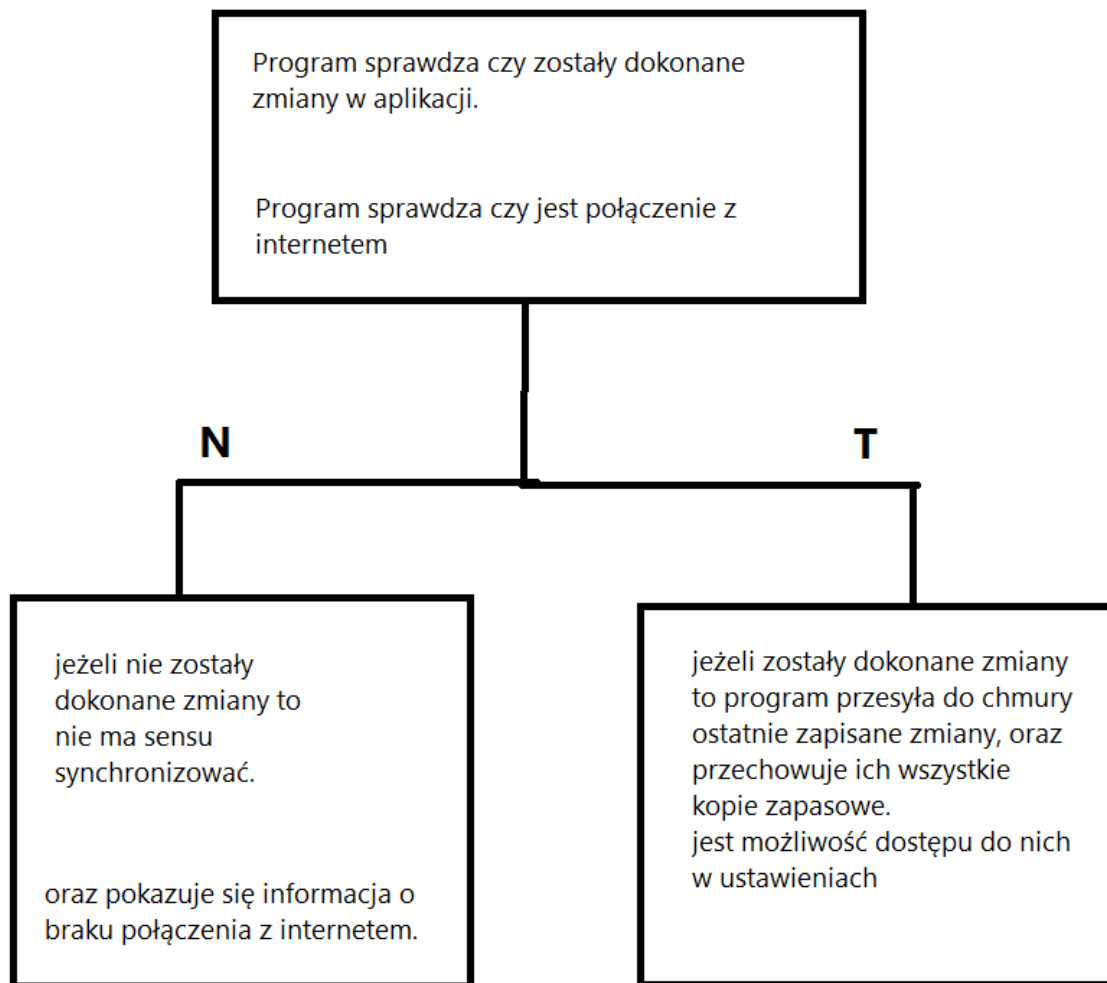
- numer – 6
- nazwa – Możliwość tworzenia własnych narzędzi typu follow Focus, jeździec i sparowania go z programem
- projekt graficzny:



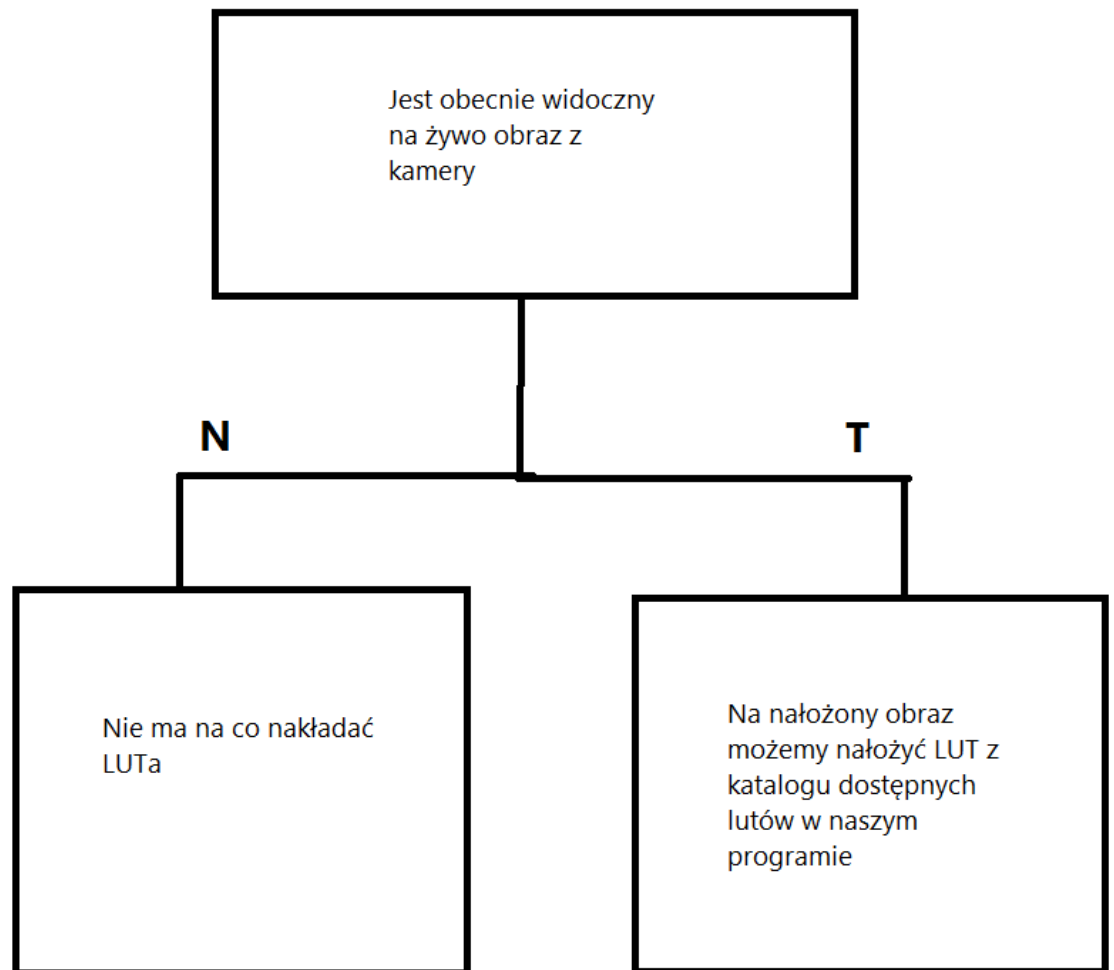
- numer – 7
- nazwa – Możliwość streamowania obrazu z kamery na żywo do telewizji, lub różnych internetowych serwisów (typu YouTube, Twitch etc.)
- projekt graficzny:



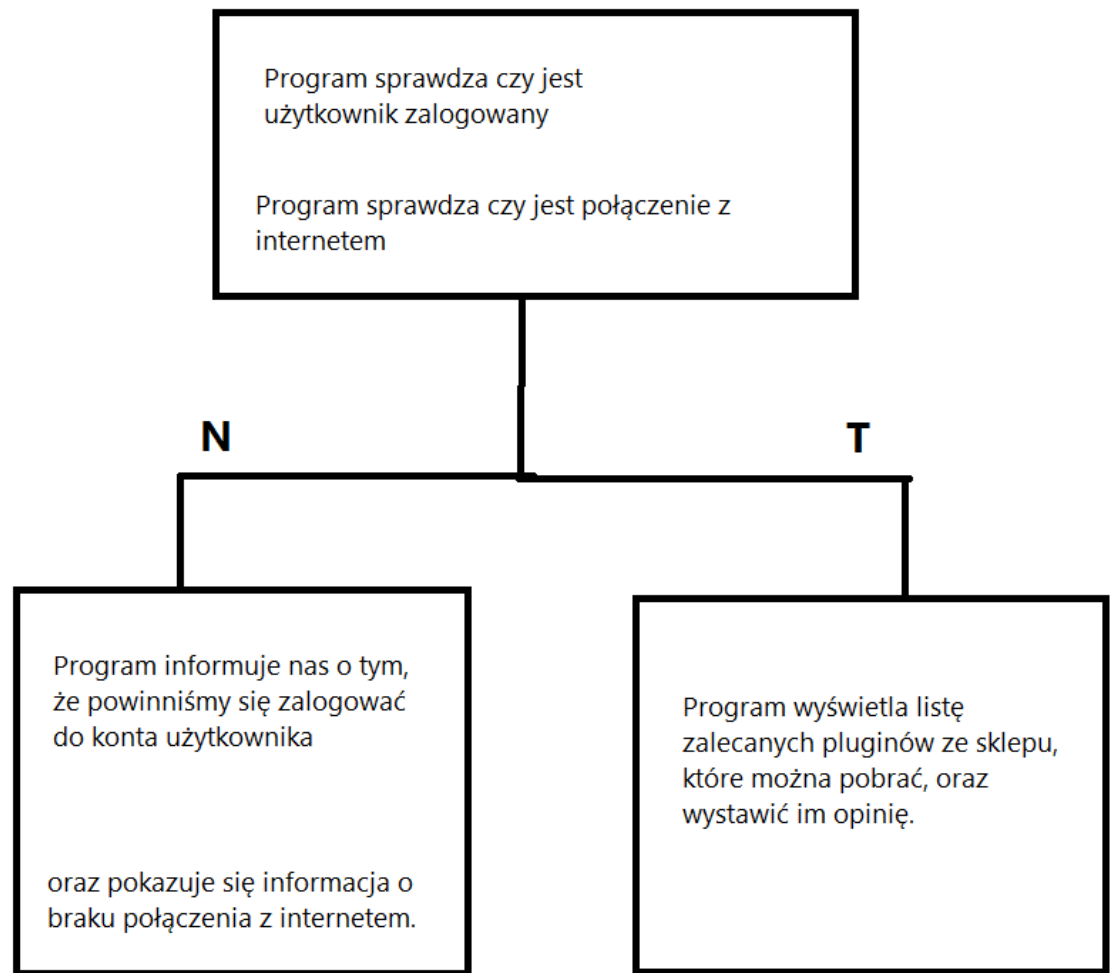
- numer – 8
- nazwa – Synchronizacja ustawień do chmury przy wykupieniu odpowiedniej subskrypcji przez klienta
- projekt graficzny:



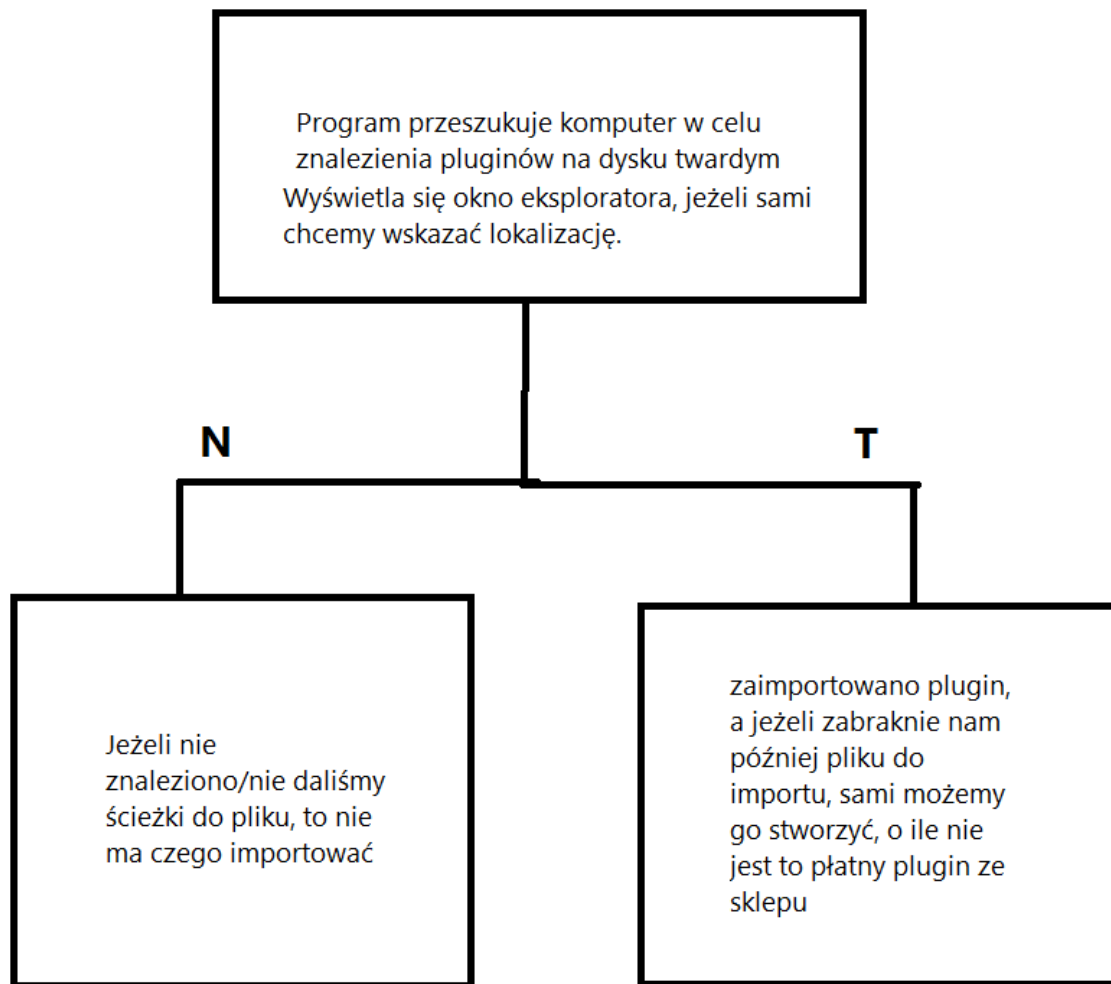
- numer – 9
- nazwa – Możliwość nakładania LUTów na podglądany obraz z kamery
- projekt graficzny:



- numer – 10
- nazwa – Narzędzie posiada listę zalecanych pluginów, oraz wbudowany sklep z nimi, wraz z opiniami innych użytkowników.
- projekt graficzny:



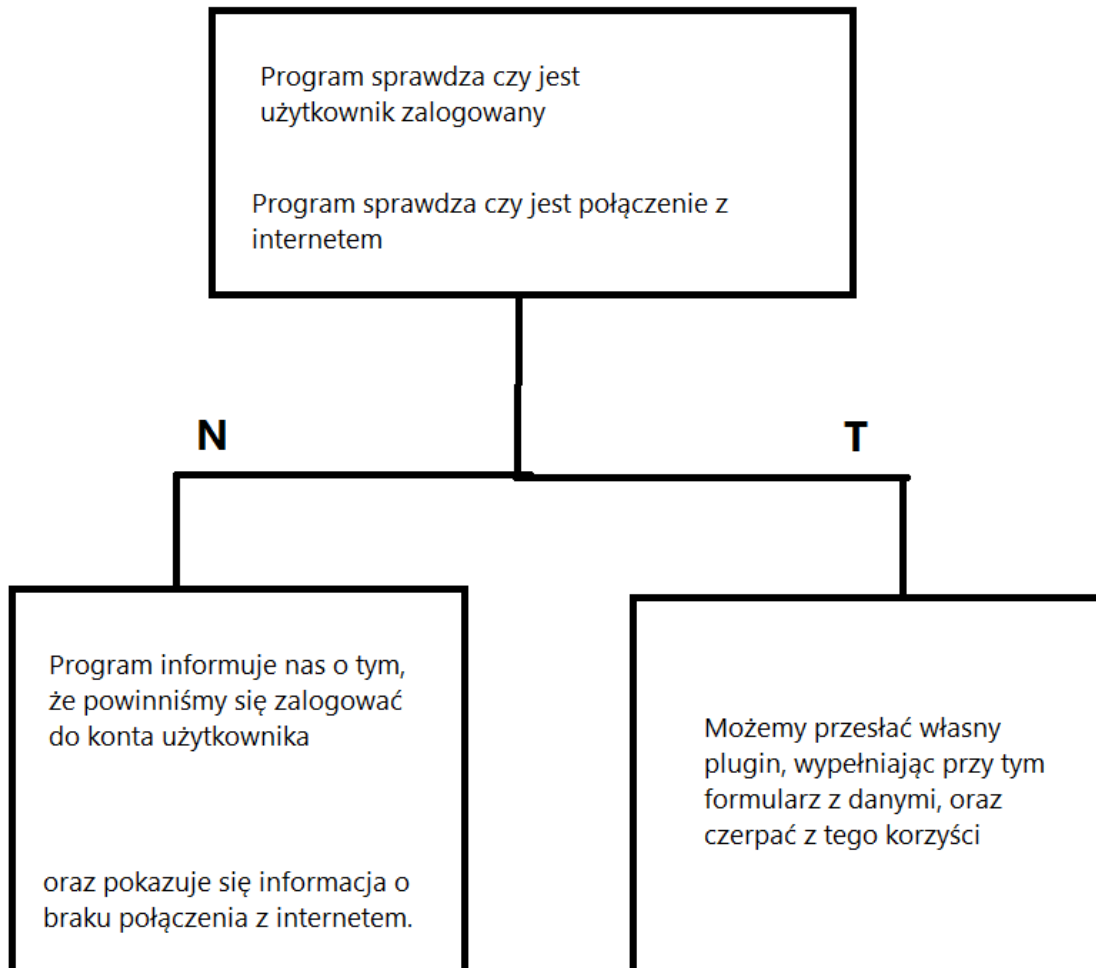
- numer – 11
- nazwa – Możliwość importu pluginów z jakichkolwiek innych źródeł za pomocą pliku, jak i jego eksportu z programu.
- projekt graficzny:



- numer – 12
- nazwa – Możliwość przesłania własnego pluginu do weryfikacji przez twórców/administrację pod kątem awaryjności/działania/braku zagrożeń i

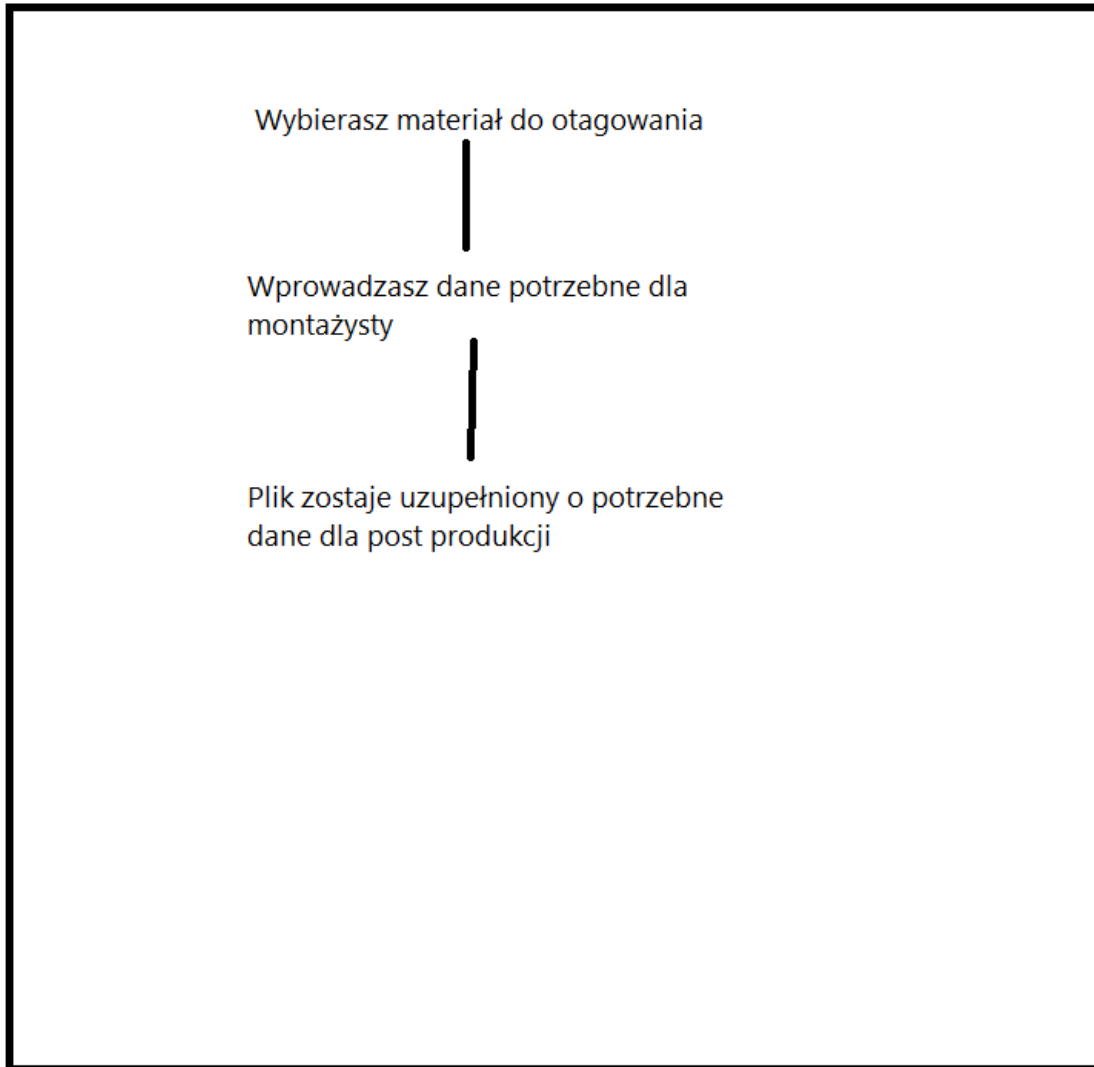
możliwość wrzucenia tego do sklepu, oraz czerpania z tego korzyści (70% dla twórców pluginu, 30% dla twórców aplikacji)

- projekt graficzny:

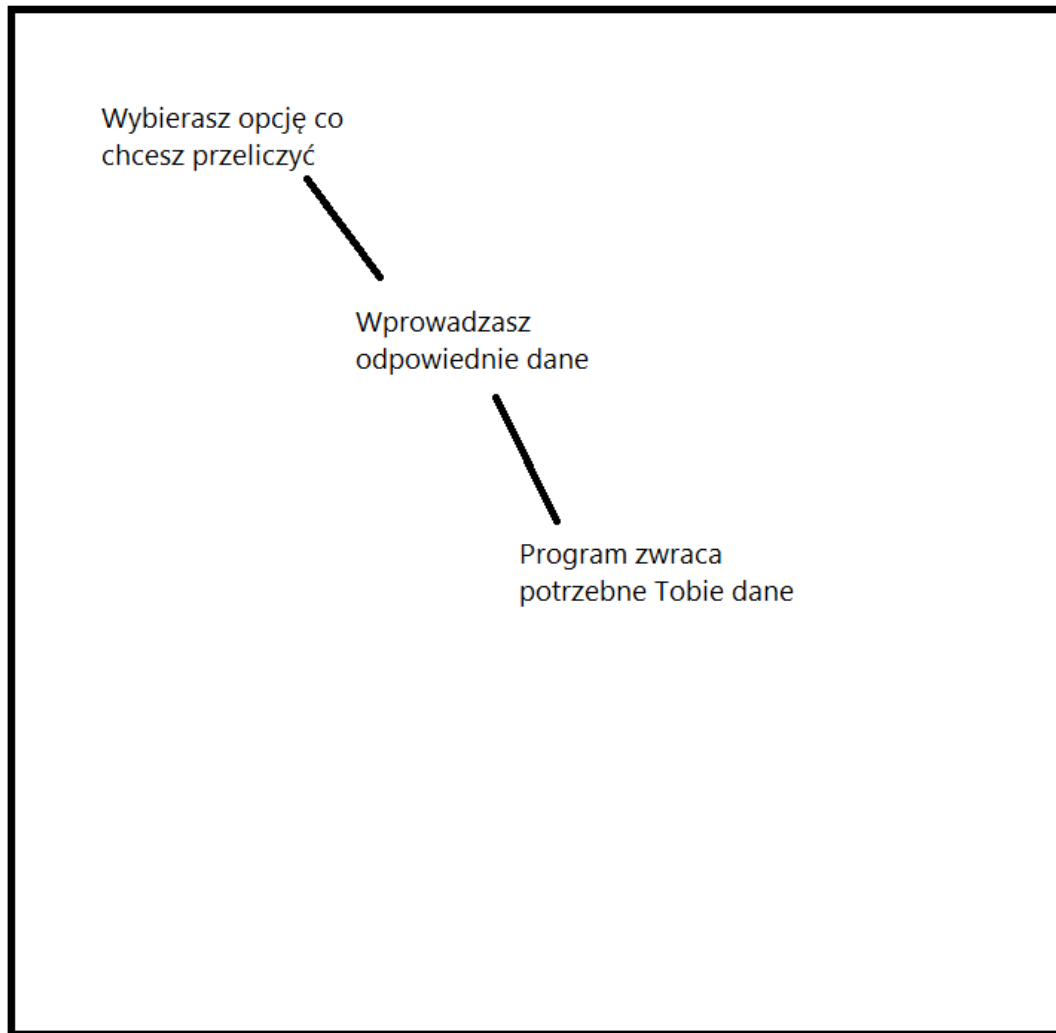


- numer – 13
- nazwa – Możliwość nakładania na nagrywany materiał informacji potrzebnych dla montażysty (scena, ujęcie, etc.)

- projekt graficzny:



- numer – 14
- nazwa – Wbudowane narzędzia do kalkulacji crop factor matrycy przy wybraniu odpowiedniej rozdzielczości/proporcji obrazu, kalkulacji czasu migawki/shutter angle (aby uzyskać obraz bez migotania w danej części świata (60Hz w PA, 50Hz w Europie i przynajmniej większej części Azji), kalkulacji ekspozycji (parametry wejściowe: shutter angle, filtr nd, frame rate, transmisja światła obiektywu)
- projekt graficzny:



9.7 Procedura wdrożenia

jeśli informacje w harmonogramie nie są wystarczające (a zapewne nie są)

10 Dokumentacja dla użytkownika

Opcjonalnie – dla chętnych

Na podstawie projektu docelowej aplikacji, a nie zaimplementowanego prototypu architektury

4-6 stron z obrazkami (np. zrzuty ekranowe, polecenia do wpisania na konsoli, itp.)

- pisana językiem odpowiednim do grupy odbiorców – czyli najczęściej nie do informatyków
- może to być przebieg krok po kroku obsługi jednej głównej funkcji systemu, kilku mniejszych, instrukcja instalacji lub innej pomocniczej czynności.

11 Podsumowanie

11.1 Szczegółowe nakłady projektowe członków zespołu

tabela (kolumny to osoby, wiersze to działania) pokazująca, kto ile czasu poświęcił na projekt oraz procentowy udział każdej osoby w danym zadaniu oraz wiersz podsumowania – udział każdej osoby w skali całego projektu

Zadanie	Jan Biskupski	Jakub Wojciechowski
3	100	0
4.1	0	100
4.2	100	0
4.3	0	100
4.4	100	0
5.1	0	100
5.2	100	0
5.3	0	100
6.1	100	0
6.2	0	100
6.3	100	0
7.1	100	0
7.2	0	100
7.3	100	0
8.1	71	29
9.1	0	100
9.2	100	0
9.3	0	100
9.4	-	-
9.5	0	100
9.6	100	0
9.7	-	-
10	-	-
11.1	-	-

Harmonogram: (niebieskie – Jakub Wojciechowski, czerwone – Jan Biskupski)

lab2: 09.03.2021

lab3: 16.03.2021 - Wprowadzenie (4.1, 4.2, 4.3, 4.4), Specyfikacja wymagań (5.1)

lab4: 23.03.2021 - Specyfikacja wymagań (5.2, 5.3), Zarządzanie projektem (6.1, 6.2, 6.3)

lab5: 30.03.2021 - Zarządzanie ryzykiem (7.1, 7.2, 7.3)

lab6: 31.03.2021 - Zarządzanie jakością (8.1 – praca 50/50)

lab7: 13.04.2021 - -----||-----

lab8: 20.04.2021 - Projekt techniczny (9) (9.1, 9.2, 9.3, 9.4, 9.5, 9.6, 9.7) - Diagramy do projektu technicznego.

lab9: 27.04.2021 - -----||-----

lab10: 11.05.2021 - Dokumentacja dla użytkownika (10), Słownik pojęć (3), i reszta

lab11: 18.05.2021 - -----||-----, Szczegółowe nakłady projektowe członków zespołu (11.1)

lab12: 25.05.202