Mikołaj Sulkowski, Damian Pyrcz, Marcel Raś, Łukasz Radecki, Leon Sadło

**PROJEKT SYSTEMU INFORMATYCZNEGO Plagium AI**

***1. Faza strategiczna***

1. ***Opis* organizacji**

Firma specjalizuje się w organizowaniu konkursów algorytmicznych online, przyciągając programistów z całego świata. Działa na rynku od kilkunastu lat, budując reputację w branży. Konkursy odbywają się za pośrednictwem strony internetowej, na której użytkownik musi posiadać konto personalne. Na stronie wyświetlane są informacje o zbliżających się konkursach, na które trzeba się wcześniej zapisać oraz wykonać opłatę wpisową. Konkursy odbywają się średnio 52 razy w roku. Oprócz tego użytkownik ma możliwość wykupienia miesięcznej subskrypcji, która umożliwia dostęp do narzędzi oraz kursów pozwalających rozwijać swoje umiejętności. Jednym z głównych źródeł dochodów są sponsorzy oraz kontrakty z firmami informatycznymi. W czasie konkursu na stronie wyświetlane są zadania do zrealizowania i pole do odsyłania działającego kodu.

Mimo doświadczenia, firma nie posiada zaawansowanych narzędzi informatycznych do wykrywania plagiatów. Analiza kodów odbywa się głównie ręcznie lub z wykorzystaniem podstawowych narzędzi, co nieefektywnie obciąża zespół audytorów i wydłuża czas analizy. W związku z tym, planowane jest wdrożenie nowoczesnego systemu informatycznego, który w czasie rzeczywistym będzie automatycznie analizował zgłaszane kody pod kątem plagiatu. Dzięki temu proces weryfikacji stanie się nie tylko szybszy i bardziej efektywny, ale również znacznie bardziej rzetelny. Obecne problemy z niesprawiedliwością ocen, wynikające z niewystarczających narzędzi do wykrywania plagiatów, sprawiły, że firmy przestały sponsorować konkursy. Wdrożenie nowego systemu poprawi wiarygodność konkursów, co pozwoli odzyskać zaufanie sponsorów i przyciągnąć nowych partnerów, umacniając tym samym pozycję firmy na rynku.

**W skład struktury organizacyjnej wchodzą:**

**Główni organizatorzy:** Zarządzają platformą konkursową, przeglądają raporty z analizy kodu oraz ustalają zasady i komunikują się z uczestnikami. Ich rolą jest zapewnienie, że konkursy przebiegają sprawnie i zgodnie z ustalonymi zasadami.

**Audytorzy kodów:** Weryfikują zgłoszenia uczestników, oceniając je pod kątem plagiatów oraz jakości kodu. Analizują wyniki wykrytych podobieństw i podejmują decyzje o dalszych krokach w przypadku podejrzeń o nieprawidłowości.

**Programiści:** Odpowiedzialni za wsparcie techniczne oraz utrzymanie platformy konkursowej. Dbają o jej prawidłowe funkcjonowanie, wprowadzają niezbędne aktualizacje oraz usprawnienia, a także reagują na zgłaszane problemy techniczne.

**Biuro Obsługi Klienta:** Zespół, który zbiera wymagania oraz analizuje procesy, które system ma wspierać. Zapewnia optymalne doświadczenie użytkowników poprzez komunikację oraz rozwiązywanie ich problemów.

**Testerzy:** Odpowiedzialni za testowanie systemu pod kątem wydajności, funkcjonalności i poprawności działania.

1. ***Problem biznesowy***

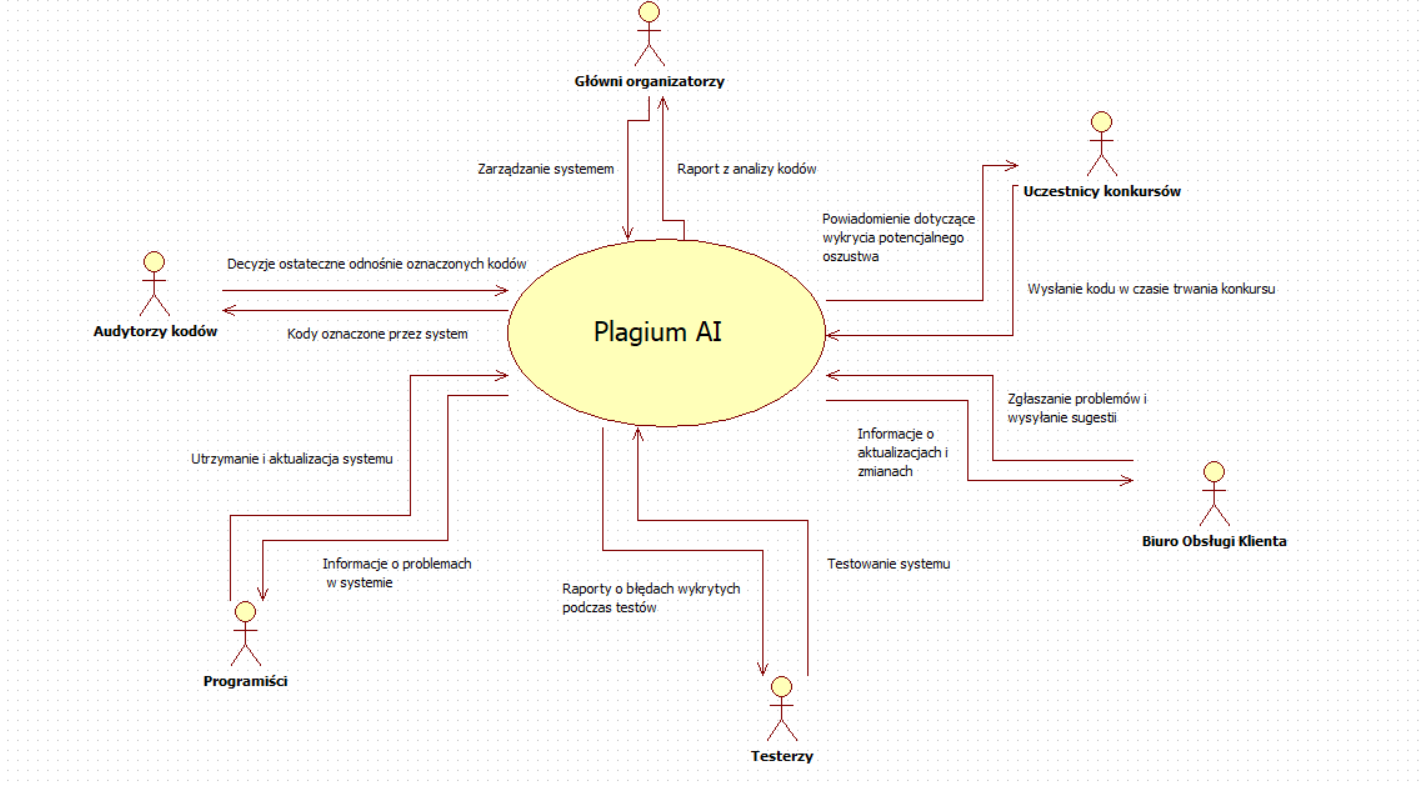
* Niewydajny system weryfikacji nie umożliwia efektywnego sprawdzania zgłaszanego kodu co wpływa na mniejszą wiarygodność wyników
* Próby tuszowania oszustwa i obejścia systemu przez część uczestników
* Długi czas oczekiwania na weryfikację kodów
* Wycofywanie się sponsorów i firm
* Mniejsza przewaga konkurencyjna wobec podobnych firm
* Malejące zainteresowanie programistów z powodu obniżonej rzetelności w konkursach
* Utrata wizerunku firmy na rynku globalnym
* W większości plagiat jest wykrywany tylko w tym języku programowania, w którym został on pierwotnie napisany.

1. ***Cel biznesowy***

* Usprawnienie pracy audytorów
* Automatyczna analiza kodów uczestników, identyfikująca plagiaty na poziomie logicznym.
* Większa sprawiedliwość w konkursach
* Przyspieszenie weryfikowania kodów uczestników
* Zwiększenie zainteresowania wśród sponsorów i firm oraz dalsza współpraca z wcześniejszymi
* Zwiększenie liczby nowych użytkowników biorących udział w konkursach
* Polepszenie i utrzymywanie wizerunku firmy na odpowiednim poziomie

1. ***Interesariusze systemu***

* Główni organizatorzy
* Uczestnicy konkursów
* Biuro Obsługi Klienta
* Testerzy
* Programiści
* Audytorzy kodów

******

1. ***Wymagania funkcjonalne systemu***

* Wykrywanie podejrzanych fragmentów kodu, które będą dodatkowo weryfikowane przez audytorów lub automatycznie poprzez oprogramowanie.
* Po wysłaniu kodu przez użytkownika, system powinien automatycznie przeprowadzać analizę pod kątem podobieństw z innymi zgłoszeniami w ramach tego samego konkursu.
* System powinien ignorować zmiany w nazwach zmiennych oraz formatowaniu kodu, koncentrując się głównie na logice programu i strukturze algorytmów.
* Umożliwienie definiowanie punktu startowego analizy, co pozwoli na eliminację zbędnych fragmentów kodu.
* Generowanie raportów z wynikami analizy, które będą dostępne dla głównych organizatorów i audytorów kodu.
* Obsługiwanie wiele języków programowania, dostosowując analizę do różnych formatów zgłoszeń.
* System powinien przechowywać i przetwarzać identyfikatory kont autorów kodów oraz powiązane z nimi zgłoszenia konkursowe.
* Wymagane jest automatyczne grupowanie zgłoszeń na podstawie konkursu i zadania, co pozwala na precyzyjną analizę porównawczą.
* System musi analizować kod źródłowy pod kątem podobieństw strukturalnych między zgłoszeniami z tego samego konkursu.
* System powinien umożliwiać organizatorom definiowanie procentowego poziomu podobieństwa kodu, powyżej którego zgłoszenie będzie oznaczane jako potencjalny plagiat

1. ***Wymagania niefunkcjonalne systemu***

* **Zdatność pielęgnacji:** Oprogramowanie powinno być łatwe w utrzymaniu i aktualizacji
* **Dokładność:** System powinien zapewniać wysoką precyzję wykrywania plagiatu, minimalizując fałszywe pozytywy i negatywy.
* **Przenośność:** Oprogramowanie powinno być zaprojektowane z myślą o łatwej migracji na różne platformy sprzętowe oraz systemy operacyjne.
* **Bezpieczeństwo:** System musi chronić dane uczestników przed nieautoryzowanym dostępem i zapewniać bezpieczną wymianę informacji.
* **Czytelność i Intuicyjność:** System powinien mieć na tyle prosty i intuicyjny interfejs by nie sprawiać problemów nawet podczas pierwszego użytkowania.
* **Wydajność:** System powinien efektywnie analizować przesłany kod pod kątem plagiatu, a pełny raport z oceną powinien być generowany w ciągu 24 godzin od zakończenia konkursu, zapewniając użytkownikom szybki dostęp do wyników.

1. ***Ograniczenia***

* **Czas:** Oprogramowanie musi być gotowe w ciągu roku
* **Budżet:** 770 000 PLN
* Rozwój systemu i sztucznej inteligencji 450 000 PLN
* Opracowanie algorytmów antyplagiatowych opartych na AI
* Integracja AI z bazą danych i systemem backendowym
* Implementacja funkcji systemu, takich jak analiza tekstu, porównywanie kodów
* Infrastruktura techniczna 50 000 PLN
* Koszty serwerów, hostingu i zasobów obliczeniowych potrzebnych do trenowania modeli AI
* Oprogramowanie i narzędzia niezbędne do rozwoju i utrzymania systemu
* Testowanie i optymalizacja 70 000 PLN
* Przeprowadzenie testów jednostkowych i integracyjnych
* Testy beta z udziałem użytkowników, weryfikacja dokładności AI w wykrywaniu plagiatów
* Optymalizacja algorytmów na podstawie wyników testów
* Zarządzanie projektem i koordynacja 90 000 PLN
* Planowanie, nadzór nad realizacją projektu
* Koordynacja pracy nad różnymi etapami projektu, w tym rozwojem, testowaniem i wdrażaniem
* Szkolenia i wsparcie 40 000 PLN
* Szkolenia dla użytkowników końcowych systemu
* Wsparcie techniczne po wdrożeniu
* Rezerwa na nieprzewidziane wydatki 70 000 PLN

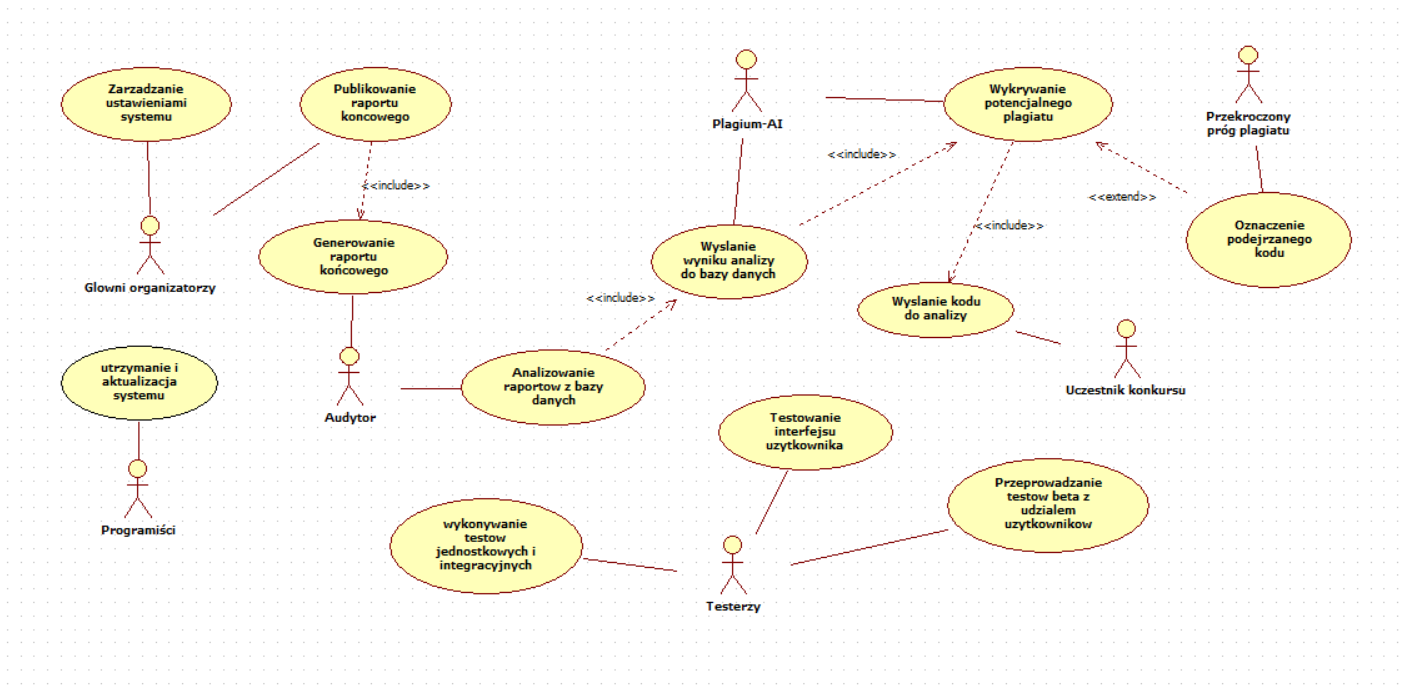
***h.*** **Harmonogram:**

* Faza planowania i analizy wymagań (1 miesiąc)
  + Określenie szczegółowych wymagań
  + Przeprowadzenie analizy konkurencji
  + Zidentyfikowanie danych dostarczonych przez firmę do trenowania AI
* Faza projektowania (1 miesiąc)
  + Tworzenie architektury systemu
  + Projektowanie interfejsu użytkownika
  + Przygotowanie infrastruktury do przetwarzania danych
* Faza rozwoju (5 miesięcy)
  + Implementacja podstawowych funkcji systemu
  + Integracja algorytmów antyplagiatowych opartych na AI
  + Trenowanie modeli AI na danych dostarczonych przez firmę
  + Tworzenie bazy danych i logiki backendowej
* Faza testowania (2 miesiące)
  + Testy jednostkowe i integracyjne
  + Beta testing z udziałem użytkowników
  + Optymalizacja algorytmów AI na podstawie wyników testów
* Faza wdrożenia (1 miesiąc)
  + Wdrożenie systemu na serwerze produkcyjnym
  + Szkolenie użytkowników i zapewnienie wsparcia technicznego
* Faza monitorowania i utrzymania (2 miesiące)
  + Monitorowanie działania systemu
  + Wprowadzanie poprawek i aktualizacji na podstawie opinii użytkowników
  + Dalsze szkolenia i rozwój funkcji systemu

***i. Zespół projektowy***

* **Kierownik projektu:** Koordynuje pracami całego zespołu.
* **Architekt systemu:** Projektuje architekturę systemu oraz integruje ją z obecną infrastrukturą.
* **Programiści:** Implementują system, są odpowiedzialni za dalszy rozwój systemu.
* **Testerzy:** Są odpowiedzialni za testowanie stabilności systemu, wykrywanie błędów i obejść zabezpieczeń.
* **Specjaliści ds. bezpieczeństwa:** Zapewniają, że system jest zgodny z przepisami dot. ochrony danych.
* **Specjalista UX/UI:** Zaprojektowanie przyjaznego w obsłudze interfejsu użytkownika

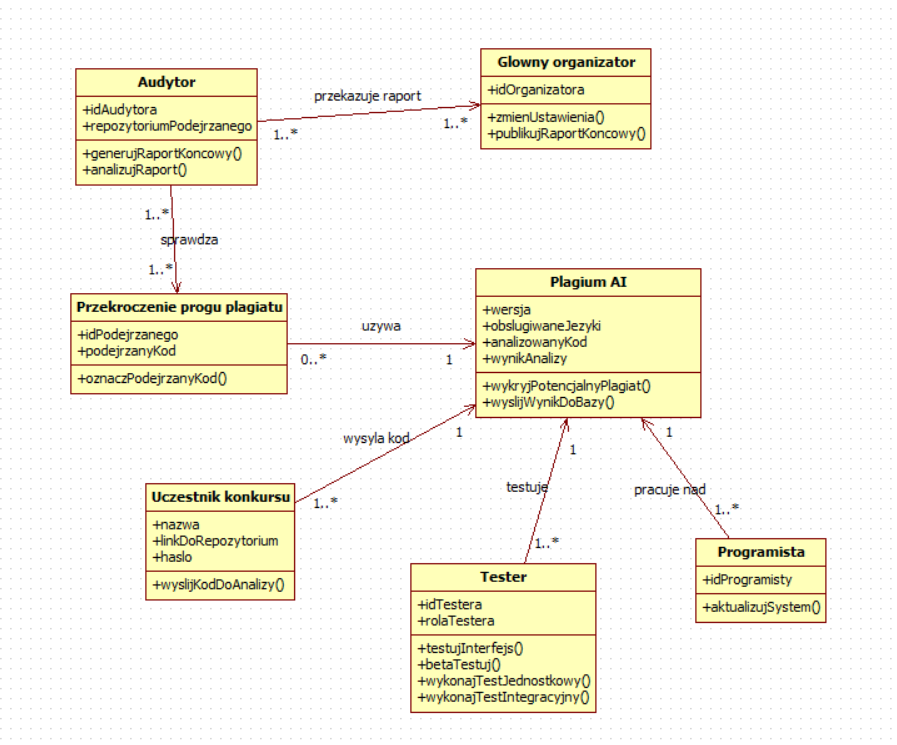
***2.Diagram przypadków użycia***

******

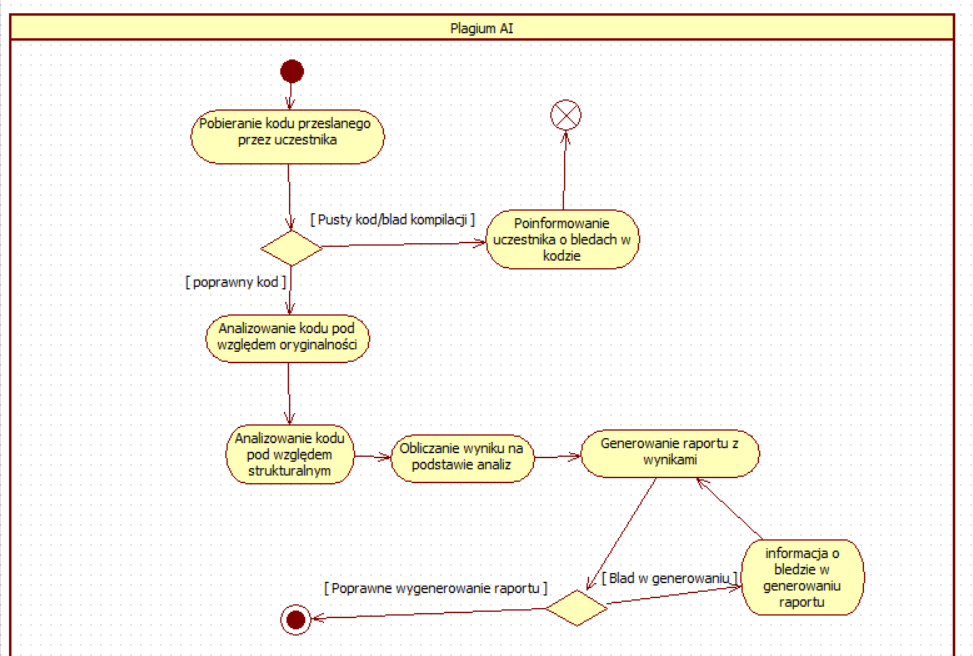
***3.Dokumentacja przypadku użycia***

|  |  |
| --- | --- |
| **Nazwa:** | Wykrywanie potencjalnego plagiatu |
| **Numer:** | 001 |
| **Twórca:** | Mikołaj Sulkowski, Damian Pyrcz, Marcel Raś, Łukasz Radecki, Leon Sadło |
| **Poziom ważności:** | Określenie poziomu ważności przypadku z perspektywy  użytkownika: **wysoki** |
| **Typ przypadku użycia:** | Określenie typu przypadku użycia z punktu widzenia jego  złożoności oraz ważności dla zaspokojenia potrzeb użytkownika:  **ogólny; niezbędny** |
| **Aktorzy:** | Uczestnik konkursu, Plagium AI |
| **Krótki opis:** | Analizuje kod pod kątem plagiatu |
| **Warunki wstępne:** | Wysłanie kodu do analizy |
| **Warunki końcowe:** | Wysłanie wyniki analizy do bazy danych |
| **Główny przepływ zdarzeń:** | 1.System rejestruje wysłanie kodu przez uczestnika  2.System analizuje kod pod względem oryginalności na podstawie wcześniejszych prac i danych z Internetu  3.System analizuje kod pod względem strukturalnym, na podstawie wzorców internetowych o popularnych schematach  4.System na podstawie analiz szacuje możliwość wystąpienia plagiatu w procentach  5.System generuje raport o wyniku analizy |
| **Alternatywne przepływy zdarzeń** | 1a.Wysłany kod jest pusty  1b.Wysłany kod zawiera błędy (Nie można skompilować)  5a.Błąd w generowaniu raportu |
| **Specjalne wymagania:** | System ma być szybki, dokładny, łatwy w implementacji i konserwacji |
| **Notatki i kwestie:** | brak |

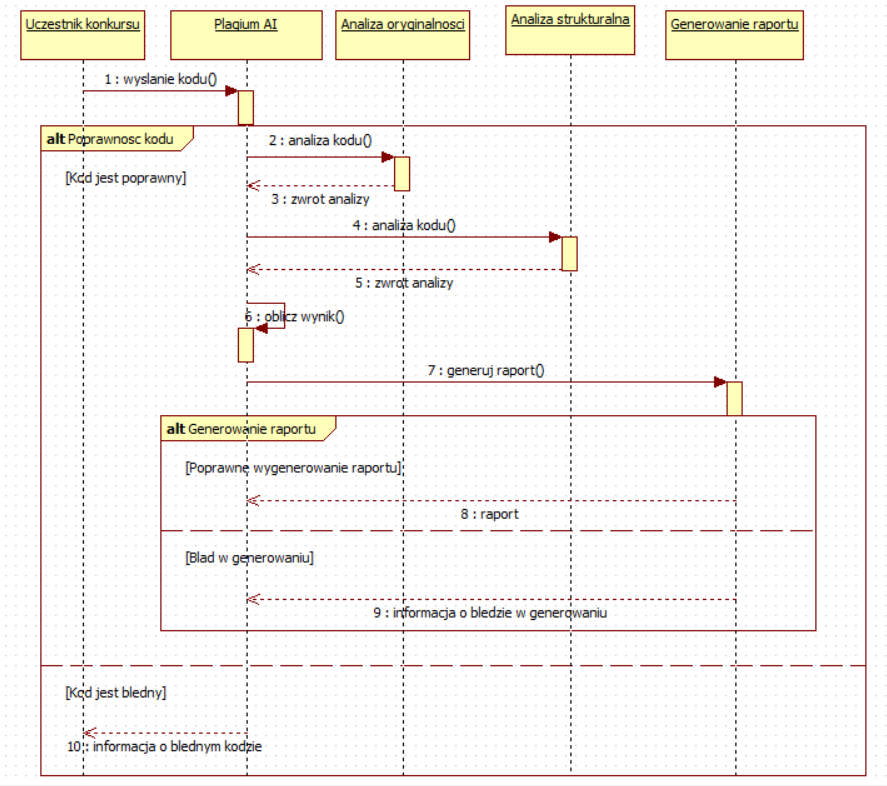
***4.Diagram Klas***

******

***5.Diagram Czynności***

******

***6.Diagram Sekwencji***

******

***7.Formatka Wejściowa***

******

**Przykładowe dane:**

nickname: nickNickowskyy0

password: adminAdmin123\_

repo link: https://github.com/dev1/repo

***8.Formatka Wyjściowa***

******