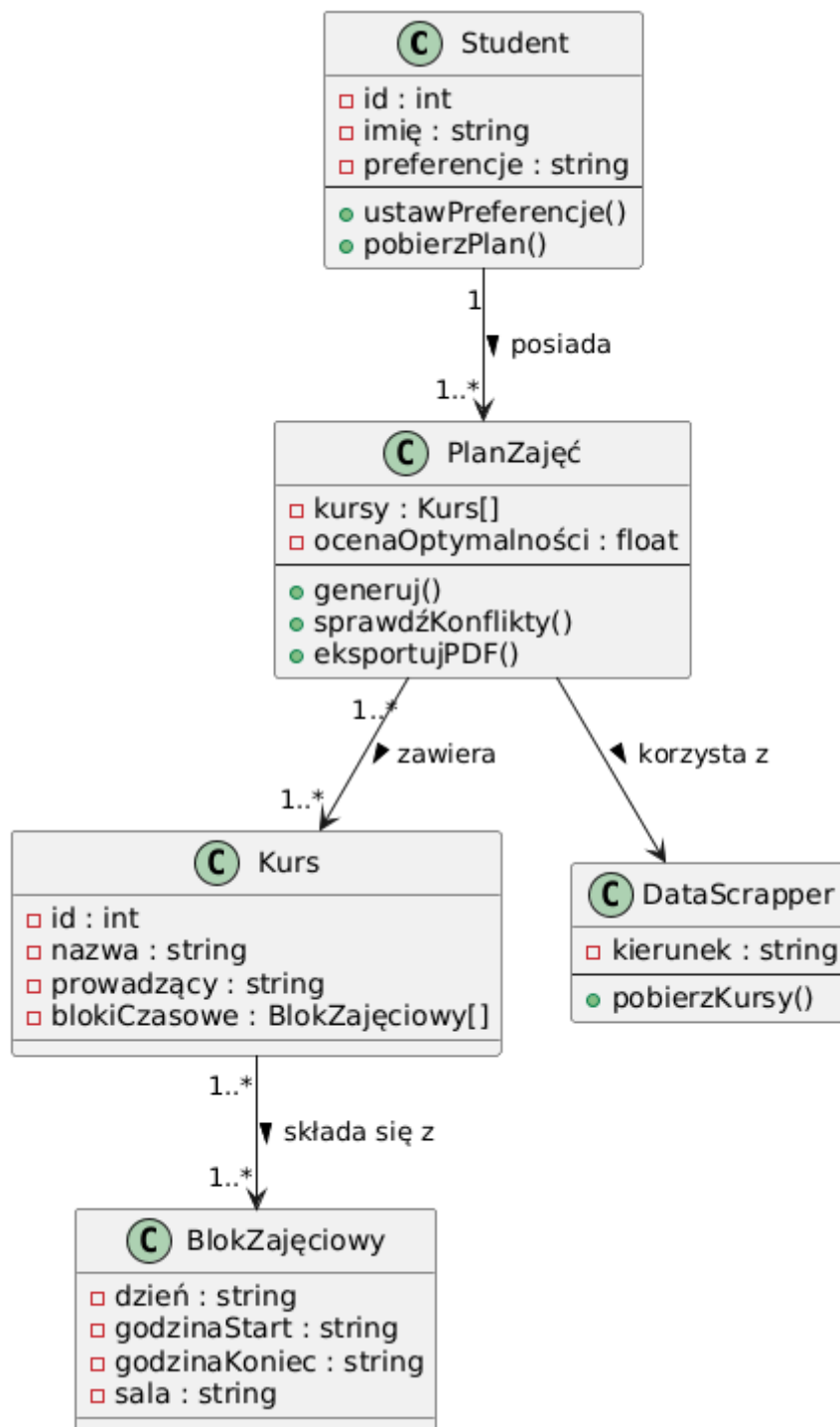


## Wyszczególnienie ryzyk projektowych – Projekt MUGA

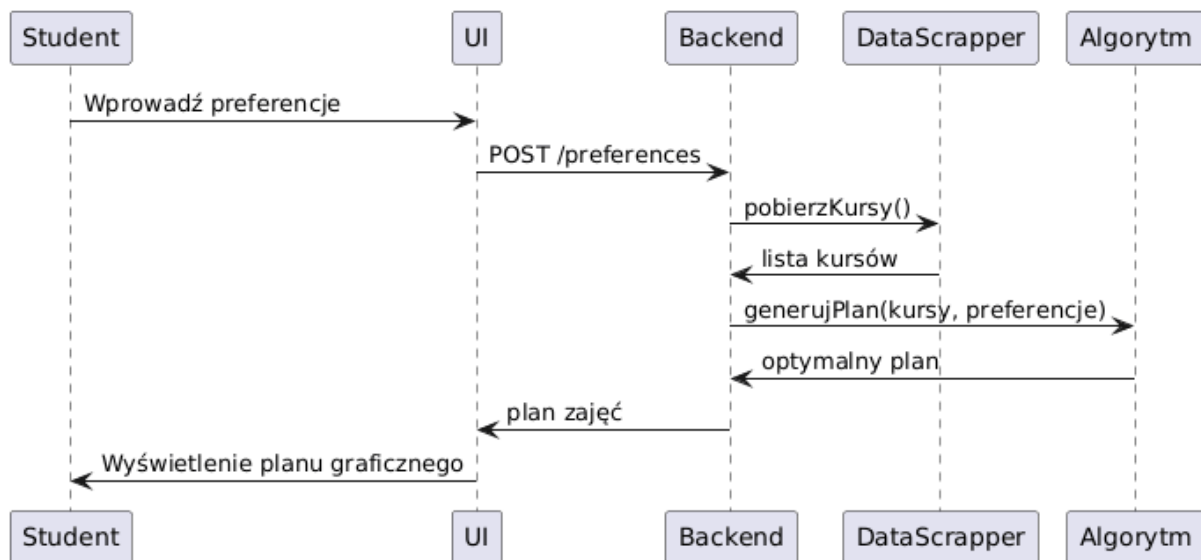
Opis ryzyka	Prawdopodobieństwo	Poziom ryzyka	Plan naprawczy (reakcja)	Metody zapobiegania
Problemy z automatycznym pobieraniem danych z portalu uczelni (zmiana struktury strony, CAPTCHA, logowanie)	Wysokie	Krytyczne	Możliwość importu danych ręcznie (np. wcześniej stworzonej bazy danych).	Playwright z mechanizmem wykrywania zmian w strukturze DOM; testy regresyjne po aktualizacjach.
Błędy w algorytmie generowania planu (niewłaściwe rozpoznanie kolizji lub preferencji)	Niskie	Wysokie	Analiza i poprawa logiki algorytmu; dodanie testów jednostkowych.	Regularne testy jednostkowe i integracyjne, recenzje kodu, testy z danymi przykładowymi.
Niewystarczająca wydajność przy dużej liczbie kursów i studentów	Średnie	Średnie	Optymalizacja algorytmu i zapytań, wprowadzenie cache danych.	Analiza wydajności, wczesna optymalizacja kluczowych fragmentów kodu.
Niepowodzenie integracji frontend-backend (problemy z API)	Średnie	Średnie	Diagnostyka komunikacji, fallback do wersji offline z przykładowymi danymi.	Stosowanie wspólnej dokumentacji, testy integracyjne.
Brak akceptacji interfejsu użytkownika (nieintuicyjna obsługa)	Średnie	Średnie	Przeprowadzenie testów z użytkownikami i szybka iteracja UI/UX.	Włączenie użytkowników w proces testowania od wczesnego etapu projektu.
Opóźnienia w harmonogramie (np. przez integracje lub testy)	Wysokie	Wysokie	Przesunięcie mniej krytycznych funkcji na kolejne sprinty; zwiększenie zasobów czasowych.	Wczesne planowanie kamieni milowych, monitorowanie postępu (np. Trello).
Awaria środowiska produkcyjnego lub serwera	Niskie	Średnie	Przeniesienie usług na zapasowe środowisko.	Wykorzystanie chmury z wysoką dostępnością, monitoring stanu serwera.

# Diagramy UML:

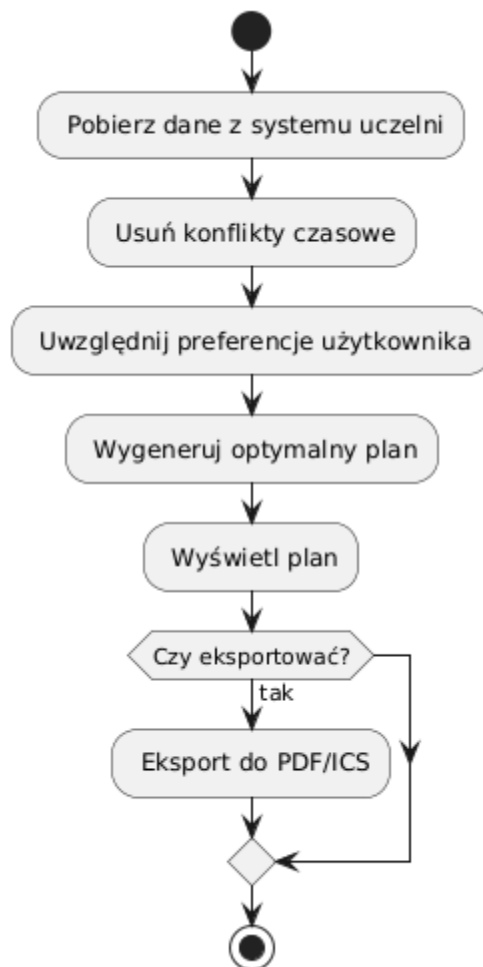
## 1) Class diagram



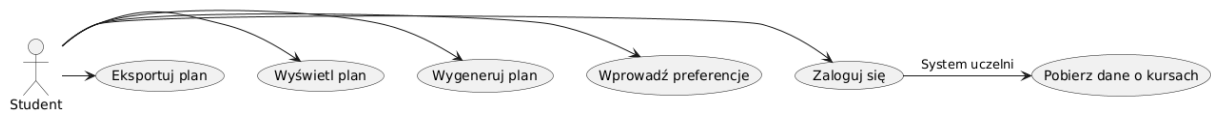
### Sequence diagram:



### Activity diagram:



### Use case diagram:



### Podział pracy na 3-tygodniowe iteracje:

- 1) Stworzenie DataScraper'a (Playwright) i algorytmu do układania planu oraz ich integracja przez backend.
- 2) Stworzenie czytelnego UI i jego integracja z działającym backend'em oraz przeprowadzenie alpha i beta testów systemu. Wprowadzenie końcowych poprawek oraz dokumentacja projektu.