


Numer zlecenia oraz nazwa i akronim projektu: <i>[numer projektu nie jest jeszcze nadany] EduPlaner CzerwMaKonBas</i>	Zleceniodawca: <i>Polsko-Japońska Akademia Technik Komputerowych</i>	Zleceniobiorca: 
Zespół projektowy: <i>Czerwiński Cyprian Koniak Kamil Małolepszy Michał Bastek Aleksander</i>	Kierownik projektu: <i>Czerwiński Cyprian</i>	
Nazwa dokumentu: <i>Specyfikacja Wymagań Systemowych</i>	Odpowiedzialny za dokument: <i>Bastek Aleksander</i>	Opiekun projektu: <i>Marta Łabuda</i>

Historia dokumentu

Wersja	Opis modyfikacji	Rozdział / strona	Autor modyfikacji	Data
1.0	Wstępna wersja	całość	Cały zespół projektowy	23.10.2024 r.
1.1	Recenzja i uzupełnienie	całość	Cały zespół projektowy	30.10.2024 r.

Opis etapu: Celem etapu jest udokumentowanie procesu i podjętych decyzji związanych ze zbieraniem, przetwarzaniem i analizowaniem wymagań.

Oczekiwane produkty: Specyfikacja wymagań systemowych w postaci dokumentu o strukturze według poniższego szablonu i nazewnictwie zgodnym ze składnią **SWS-GrA(XYZ)_v0** – gdzie A to numer porządkowy grupy; X, Y, Z to nazwiska poszczególnych członków, a 0 to numer wersji.

1.1 Cel dokumentu

{Określenie celu, jakiemu ma służyć ten dokument; tutaj: zdefiniowanie wymagań na podstawie analizy otoczenia projektu / przedsięwzięcia oraz analizy potrzeb klienta}

Dokument stanowi formalne przedstawienie wymagań systemowych opracowanych na podstawie szczegółowej analizy potrzeb użytkowników oraz otoczenia biznesowego projektu EduPlaner. Jego celem jest wsparcie interesariuszy w zrozumieniu projektu oraz zapewnienie precyzyjnej definicji wymagań EduPlanera, uwzględniającej analizę kontekstu, potrzeb oraz warunków technicznych niezbędnych do realizacji.

1.2 Zakres dokumentu

{Określenie, co zawiera dokument; tutaj: analiza otoczenia – klienci, kontekst biznesowy systemu, określenie udziałowców, zdefiniowanie wymagań}

Dokument obejmuje szczegółową analizę otoczenia projektu EduPlaner, uwzględniającą identyfikację kluczowych grup docelowych, specyficznych potrzeb klientów oraz analizę kontekstu biznesowego, w którym system będzie funkcjonował. Zakres dokumentu obejmuje również określenie udziałowców i ich oczekiwań, co umożliwia precyzyjne dopasowanie funkcji systemu do rzeczywistych wymagań biznesowych i organizacyjnych.

Dokument zawiera także kompletne zestawienie wymagań systemowych, obejmujących zarówno wymagania funkcjonalne, jak i нефункционалне. Dzięki temu dostarcza spójnej podstawy do dalszych etapów realizacji projektu, umożliwiając ich systematyczną weryfikację na wszystkich etapach cyklu życia aplikacji.

1.3 Dokumenty powiązane

{Pełna lista opracowań poza zakresem dokumentu, lecz powiązanych merytorycznie z tym dokumentem; ujęte wersje poszczególnych dokumentów; także poprzednie wersje dokumentu SWS}

*Dokument Założeń Wstępnych
Karta Projektu wraz z załącznikami:*

- elevator pitch,
- rich picture,
- prezentacja projektu.

1.4 Odbiorcy

- *{Określenie adresatów dokumentu np. zleceniodawca, zleceniobiorca, członkowie zespołu projektowego oraz (jeśli znane) wymienione z nazwiska osoby, do których dokument ma dotrzeć; aspekty prawne i licencyjne (kto posiada prawa autorskie do dokumentu / jaki akt prawny je reguluje?)}*

Zleceniodawca:

- **Polsko-Japońska Akademia Technik Komputerowych (PJATK)** – Instytucja edukacyjna zlecająca projekt.

Zleceniobiorca:

- **Zespół projektowy** – Odpowiedzialny za realizację projektu EduPlaner, w skład którego wchodzi Cyprian Czerwiński, Kamil Koniak, Michał Małolepszy, Aleksander Bastek.

- **Użytkownicy systemu:**

- **Administratorzy** – Osoby odpowiedzialne za tworzenie i edycję planów zajęć, mogą to być członkowie administracji uczelni lub wyznaczeni planiści. Maksymalna liczba administratorów to 10.
- **Nauczyciele** – Prowadzą zajęcia na uczelni. Mają dostęp do odczytu planu lekcji oraz swojego grafiku. Mogą również komunikować się ze studentami. Maksymalna liczba nauczycieli to 100.
- **Studenci** – Uczestnicy procesu kształcenia. Mają dostęp do przeglądu swojego planu lekcji oraz możliwość otrzymywania powiadomień od nauczycieli. Maksymalna liczba studentów to 500.
- **Klienci wewnętrzni:**
- **Opiekun projektu** – Marta Łabuda.
- **Zespół projektowy (Aleksander Bastek, Cyprian Czerwiński, Michał Małolepszy, Kamil Koniak)** – Osoby odpowiedzialne za wdrożenie aplikacji.
- **Klienci zewnętrzni:**
- **Rektor i wicerektorzy uczelni** – Przedstawiciele PJATK, odpowiedzialni za nadzór nad realizacją projektu i akceptację planów.
- **Planiści** – Osoby zewnętrzne, które mogą być zleceniobiorcami odpowiedzialnymi za tworzenie planów zajęć.
- **Personel techniczny PJATK** – Odpowiedzialny za wsparcie techniczne.
- **Inne zainteresowane strony:**
- **Administracja uczelni** – Odpowiedzialna za organizację pracy uczelni, współpracuje z zespołem projektowym.
- **Prawo o szkolnictwie wyższym i nauce** – Regulacje, które wpływają na realizację projektu, np. określenie maksymalnych godzin zajęć w tygodniu.

1.5 Słownik pojęć

{Wyjaśnienie używanych w dokumencie oznaczeń, pojęć i skrótów np. akronimy, nazwy skrótowe, itp; definicje są konieczne, aby wszyscy czytelnicy mogli zrozumieć treść dokumentu; ważne jest umieszczenie na tej liście pojęć o charakterze technicznym oraz pojęć dziedziny merytorycznej zastosowania aplikacji. Nie jest konieczne wyjaśnianie słów używanych powszechnie}

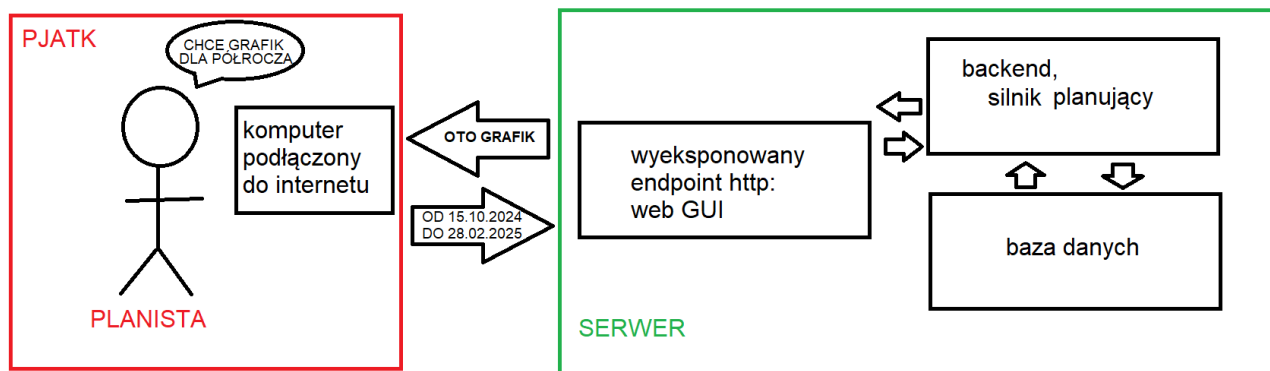
- **EduPlaner** – Projekt tworzenia systemu zarządzania harmonogramami zajęć w szkole.
- **SWS** – Specyfikacja Wymagań Systemowych; dokument zawierający szczegółowe wymagania systemu.
- **PJATK** – Polsko-Japońska Akademia Technik Komputerowych; zleceniodawca projektu.
- **Backend** – Część systemu odpowiedzialna za logikę biznesową i przetwarzanie danych, oparta na Quarkus (Java).
- **Frontend** – Interfejs użytkownika, opracowywany przy użyciu React (JavaScript), odpowiedzialny za wyświetlanie i interakcję z użytkownikiem.
- **Timefold.ai** – Biblioteka służąca do generowania harmonogramów w oparciu o algorytmy optymalizacyjne.
- **Administrator** – Użytkownik systemu z uprawnieniami do tworzenia i edycji planów zajęć.
- **Nauczyciel** – Użytkownik systemu mający dostęp do planów lekcji i możliwość komunikacji z uczniami.
- **Student** – Użytkownik systemu z dostępem do przeglądu swojego planu zajęć i powiadomień.
- **Docker** – Narzędzie do konteneryzacji aplikacji, umożliwiające oddzielne uruchamianie Frontend i Backend.
- **Zespół projektowy** – Grupa osób odpowiedzialna za realizację projektu; składa się z programistów i kierownika projektu.
- **Harmonogramowanie** – Proces tworzenia i optymalizacji planu zajęć, uwzględniający dostępność zasobów.
- **Wymagania funkcjonalne** – Wymagania definiujące, co system ma robić (np. generowanie planu lekcji).
- **Wymagania niefunkcjonalne** – Wymagania określające, jak system ma działać (np. dostępność, kompatybilność).
- **Karta Wymagania** – Sekcja dokumentu definiująca szczegółowo jedno wymaganie systemowe, jego priorytety i kryteria akceptacji.
- **MoSCoW** – Skala priorytetów dla wymagań: Must (musi), Should (powinno), Could (może), Won't (nie będzie).
- **Interfejs** – Punkt styku systemu z innymi podsystemami, umożliwiający komunikację i integrację.
- **Udziałowiec (Stakeholder)** – Osoba lub jednostka, która ma wpływ na projekt lub jest nim zainteresowana, np. administracja uczelni, studenci.
- **NF** – Wymagania niefunkcjonalne; zapewniające aspekty takie jak niezawodność, intuicyjność i kompatybilność przeglądarek.
- **Panel Zarządzania** – Moduł systemu umożliwiający administratorowi zarządzanie danymi wprowadzanymi do planu zajęć.

2 Projekt w kontekście

{Opis systemu w docelowym środowisku jego zastosowania}

2.1 Kontekst biznesowy

{Zalecane poparcie rysunkiem (np. ogólny diagram przypadków użycia) przedstawiający system w docelowym środowisku organizacyjnym i operacyjnym, zakres, integracje z innymi systemami, współdzielone bazy danych, konieczność wykorzystania szablonów, wzorców lub standardów wewnątrzorganizacyjnych; głównych udziałowców i ilustrujący relacje między nimi}



Kontekst biznesowy

System będzie składał się z dwóch głównych komponentów: części backendowej i frontendowej. Backend będzie odpowiedzialny za logikę generowania harmonogramu oraz za udostępnienie punktów końcowych (endpointów) umożliwiających interakcję użytkownika z systemem. Część backendowa i frontendowa będą uruchamiane jako odrębne kontenery w środowisku Docker, co umożliwi ich łatwe zarządzanie i skalowanie.

Backend zostanie zrealizowany w języku Java z użyciem frameworku Quarkus, który pozwala na wydajną obsługę mikrosług oraz efektywne zarządzanie zasobami. Biblioteka **Timefold.ai** zostanie wykorzystana do tworzenia harmonogramu, co umożliwi optymalizację czasu oraz elastyczne planowanie zajęć.

Frontend zostanie napisany w języku JavaScript, z użyciem biblioteki React, która pozwala na tworzenie dynamicznych i responsywnych interfejsów użytkownika, dostosowanych do potrzeb różnych grup odbiorców.

System będzie posiadał trzy role użytkowników:

1. **Administrator** – posiada możliwość tworzenia i edycji planów lekcji. Jest odpowiedzialny za wprowadzanie zmian do harmonogramu, a także zarządzanie dostępem dla nauczycieli i studentów.
2. **Nauczyciel** – ma dostęp do własnego grafiku zajęć i możliwość powiadamiania studentów o zmianach w harmonogramie.
3. **Student** – ma jedynie podgląd planu lekcji i możliwość odbierania powiadomień o ewentualnych zmianach w grafiku zajęć.

Dodatkowo, system zostanie zaprojektowany z myślą o integracji z wewnętrznymi bazami danych uczelni, co umożliwi łatwe zarządzanie danymi użytkowników oraz synchronizację z istniejącymi systemami rejestracji. System może być rozszerzalny w przyszłości o integracje z zewnętrznymi API, jednak na ten moment założono brak takiej potrzeby.

2.2 Udziałowcy

{ Udziałowiec to każdy podmiot, ożywiony bądź nie (osoba, system, urządzenie, regulacje prawne, społeczeństwo itp), który bierze udział w projekcie, lub na którego projekt może wpływać.

Dla projektów, które powstają w oparciu o istniejącą infrastrukturę techniczną, należy pamiętać o włączeniu tej infrastruktury jako udziałowca nieożywionego, którego istnienie narzuca pewne rozwiązania i wymagania}

KARTA UDZIAŁOWCA	
Identyfikator:	UOB1
Nazwa:	Administracja
Opis:	Władze uczelni wyższej. Mogą zlecać zadania podwykonawcom np. Planistom stworzenie planu.
Typ udziałowca:	Ożywiony bezpośredni
Punkt widzenia:	Jego celem jest ułatwienie zarządzania szkołą, sporządzanie i sporządzanie lepiej zoptymalizowanych harmonogramów
Ograniczenia:	Dysponuje z góry określonym ograniczonym budżetem.
Wymagania:	F01, F02, F03

KARTA UDZIAŁOWCA	
Identyfikator:	UOB2
Nazwa:	Programiści
Opis:	Zespół tworzący oprogramowanie.
Typ udziałowca:	Ożywiony Bezpośredni
Punkt widzenia:	Zależy im na tym, by oprogramowanie, które utworzą było łatwe w utrzymaniu i niezawodne. Wymagają budżetu, który pokryje koszt implementacji projektu i wynagrodzenie za ich pracę.
Ograniczenia:	Ma rolę wykonawczą nie podejmuje decyzji biznesowych, które są oparte o potrzeby, które ostatecznie definiują Władze Uczelni.
Wymagania:	F01, F02, F03, F04, F05, F06, F07

KARTA UDZIAŁOWCA	
Identyfikator:	UOB3
Nazwa:	Opiekun projektu
Opis:	Zespół tworzący oprogramowanie.
Typ udziałowca:	Ożywiony Bezpośredni
Punkt widzenia:	Zależy im na tym, by oprogramowanie, które utworzą było łatwe w utrzymaniu i niezawodne. Wymagają budżetu, który pokryje koszt implementacji projektu i wynagrodzenie za ich pracę.
Ograniczenia:	Ma rolę wykonawczą nie podejmuje decyzji biznesowych, które są oparte o potrzeby, które ostatecznie definiują Władze Uczelni.
Wymagania:	Łątu tylko symbole wymagań wyspecyfikowanych w rozdziale 3

KARTA UDZIAŁOWCA	
Identyfikator:	UOP1
Nazwa:	Nauczyciele
Opis:	Personel uczelni zajmujący się prowadzeniem zajęć.
Typ udziałowca:	Ożywiony Pośredni
Punkt widzenia:	Potrzebuje systemu, który zoptymalizuje plan do ich dostępności czasowej, stworzy harmonogram o jak najlepszej ciągłości czasowej (bez wielogodzinnych przerw między zajęciami) oraz umożliwi komunikację z uczniami.
Ograniczenia:	Nie może dowolnie wpływać na formę jaką przybierze plan. Plan nie ma być tworzony w oparciu o potrzeby niezliczone potrzeby nauczycieli. Jedyny wpływ nauczyciela na kształt planu to zadeklarowany czas dostępności na początku roku.
Wymagania:	F03, F04, F05, F06

KARTA UDZIAŁOWCA	
Identyfikator:	UOP2
Nazwa:	Studenci
Opis:	Osoby uczęszczające na zajęcia.
Typ udziałowca:	Ożywiony Pośredni
Punkt widzenia:	W ich interesie jest mieć harmonogram o jak najlepszej ciągłości godzin zajęć, bez przerw między nimi. Potrzebują dowiadywać się od nauczycieli o problemach w realizacji zajęć np. Spóźnieniach lub nieobecności wykładowcy.
Ograniczenia:	Nie powinien mieć żadnego wpływu na kształt planu. Nie jesteśmy w stanie stworzyć planu, który będzie w stanie zaspokoić potrzeby wszystkich studentów.
Wymagania	F03, F07

KARTA UDZIAŁOWCA	
Identyfikator:	UNP1
Nazwa:	Prawo o szkolnictwie wyższym i nauce
Opis:	Zbiór regulacji prawnych, którym działania uczelni są podporządkowane, również tworzenie planu.
Typ udziałowca:	Nieożywiony Bezpośredni
Punkt widzenia:	Tworząc plan lekcji powinniśmy brać pod uwagę ograniczenia prawne regulujące jego tworzenie np. Maksymalną ilość godzin nauki w tygodniu lub dopuszczalne godziny pracy uczelni.
Ograniczenia:	-
Wymagania:	F03

2.3 Klienci

{Klienci wewnętrzni są to klienci, którzy występują w ramach naszej organizacji np. nasz szef, dział finansowy, konstruktorzy, instalatorzy itp. specyfikujemy ich charakterystykę i potrzeby w odniesieniu do naszego projektu.
Klienci zewnętrzni - przedstawiciele zlecającego, którzy mogą mieć bardzo różne potrzeby np. dyrektor i administrator sieci, za klientów zewnętrznych uważa się także podwykonawców i dostawców.}

Klienci wewnętrzni:

Marta Łabuda opiekun projektu.

Aleksander Bastek, Cyprian Czerwiński, Michał Małolepszy, Kamil Koniak - Wdrożenie aplikacji.

Klienci zewnętrzni: Rektor i Wicerektorzy uczelni, planiści oraz personel techniczny PJATK.

2.4 Charakterystyka użytkowników

{Użytkownicy, ich kategorie, uprawnienia dostępu do funkcji i danych w poszczególnych trybach pracy systemu; zakładana liczebność użytkowników poszczególnych kategorii}

Administratorzy: Osoby mające uprawnienia do tworzenia planów zajęć. Mogą to być osoby wchodzące w skład władz uczelni bądź wyznaczeni do tworzenia planów planiści. Maksymalna ilość użytkowników to 10.

Nauczyciele: Prowadzą zajęcia na uczelni. Potrzebują dostęp do odczytu planu lekcji oraz planu ich własnych zajęć. Maksymalna ilość użytkowników to 100.

Studenci: Osoby uczestniczące w procesie kształcenia, wymagają dostępu do planu lekcji własnych grup. Maksymalna ilość użytkowników to 500.

3 Wymagania

{Wymaganie jest to potrzeba klienta lub ograniczenie narzucone przez innego udziałowca, wyrażone jako cecha projektowanego rozwiązania. Wymagania dzielimy na ogólne (zarówno biznesowe jak i dziedzinowe); funkcjonalne, pozafunkcjonalne; na środowisko docelowe oraz dotyczące procesu wytwarzania - odpowiednio w punktach poniżej.}

3.1 Wymagania ogólne i dziedzinowe

{Doprecyzowanie celów projektu / przedsięwzięcia zdefiniowanych na etapie zlecenia projektu / założeń wstępnych (KP, DZW). Wymagania ogólne / biznesowe / dziedzinowe pokazują system w kontekście i charakteryzują dziedzinę, w której będzie eksploatowany system, np. edukacja, finanse itp (mogą wymagać uwzględnienia specyficznych przepisów i zasad)}

KARTA WYMAGANIA			
Identyfikator:	WO1	Priorytet:	M
Nazwa	Unikanie konfliktów w przydzielaniu sal (roomConflict)		
Opis	Każda sala lekcyjna może być wykorzystywana tylko do jednej lekcji jednocześnie. System powinien identyfikować kolizje i ich unikać przy układaniu harmonogramu.		
Udziałowiec	Administracja		
Wymagania powiązane	-		

KARTA WYMAGANIA			
Identyfikator:	WO2	Priorytet:	M
Nazwa	Unikanie konfliktów w dostępności nauczycieli (teacherConflict)		
Opis	Żaden nauczyciel nie może prowadzić dwóch lekcji jednocześnie. System musi upewnić się, że każdy nauczyciel jest przypisany tylko do jednej lekcji w danym przedziale czasowym.		
Udziałowiec	Nauczyciele		
Wymagania powiązane	WO4, WO8		

KARTA WYMAGANIA			
Identyfikator:	WO3	Priorytet:	M
Nazwa	Unikanie konfliktów dla grup uczniów (studentGroupConflict)		
Opis	Każda grupa uczniów może uczestniczyć w jednych laboratoriach na raz, aby uniknąć konfliktów w harmonogramie zajęć danej grupy.		
Udziałowiec	Studenci		
Wymagania powiązane	WO6, WO7		

KARTA WYMAGANIA			
Identyfikator:	WO4	Priorytet:	S
Nazwa	Stabilność przydziału sal dla nauczycieli (teacherRoomStability)		
Opis	Nauczyciele preferują prowadzenie zajęć w tej samej sali w ciągu dnia, co pomaga zmniejszyć potrzebę przemieszczania się. System powinien penalizować sytuację, gdy nauczyciel musi przechodzić między salami.		
Udziałowiec	Nauczyciele		
Wymagania powiązane	WO2, WO5		

KARTA WYMAGANIA			
Identyfikator:	WO5	Priorytet:	S
Nazwa	Efektywność czasowa dla nauczycieli (teacherTimeEfficiency)		
Opis	Nauczyciele preferują lekcje następujące po sobie, bez zbędnych przerw. System powinien dążyć do tego, aby nauczyciele mieli zaplanowane lekcje w sposób ciągły, bez „okienek” dłuższych niż 60 minut.		
Udziałowiec	Nauczyciele		
Wymagania powiązane	WO2, WO4		

KARTA WYMAGANIA			
-----------------	--	--	--

Identyfikator:	WO6	Priorytet:	S
Nazwa	<i>Różnorodność przedmiotów dla grup uczniów (studentGroupSubjectVariety)</i>		
Opis	<i>System powinien unikać przydzielania kilku kolejnych lekcji tego samego przedmiotu, aby zapewnić większą różnorodność zajęć. System powinien penalizować sytuacje, gdy lekcje z tego samego przedmiotu są realizowane jedna po drugiej.</i>		
Udziałowiec	<i>Studenci</i>		
Wymagania powiązane	WO7		

KARTA WYMAGANIA			
Identyfikator:	WO7	Priorytet:	C
Nazwa	<i>Efektywność czasowa dla studentów (studentTimeEfficiency)</i>		
Opis	<i>Studenci również preferują lekcje następujące po sobie, bez zbędnych przerw. System powinien dążyć do tego, aby studenci mieli zaplanowane lekcje w sposób ciągły, bez „okienek” dłuższych niż 60 minut.</i>		
Udziałowiec	<i>Studenci</i>		
Wymagania powiązane	WO6		

KARTA WYMAGANIA			
Identyfikator:	WO8	Priorytet:	C
Nazwa	<i>Reakcja na zmiany (problemFactChange)</i>		
Opis	<i>System powinien umożliwiać szybkie wprowadzanie zmian w harmonogramie, na przykład w przypadku nieobecności nauczyciela lub niedostępności sali i automatycznie aktualizować harmonogramy, ograniczając zakłócenia w planie.</i>		
Udziałowiec	<i>Administracja</i>		
Wymagania powiązane	WO2		

KARTA WYMAGANIA			
Identyfikator:	WO9	Priorytet:	C
Nazwa	<i>Unikanie nadgodzin (avoidOvertime)</i>		
Opis	<i>System powinien unikać generowania nadgodzin dla nauczycieli.</i>		
Udziałowiec	<i>Prawo o szkolnictwie wyższym i nauce</i>		
Wymagania powiązane	WO5		

3.2 Wymagania funkcjonalne

{Funkcje/usługi; wymagania funkcjonalne powinny odpowiadać na pytanie: co system ma robić?

W tym podrozdziale powinny znaleźć się tabele przedstawione w podpunkcie 3.2.1 opracowane dla każdego zidentyfikowanego wymagania funkcjonalnego.}

3.2.1 Nazwa funkcji / usługi}

KARTA WYMAGANIA			
Identyfikator:	F01	Priorytet:	M
Nazwa	Panel Zarządzania		
Opis	<ul style="list-style-type: none"> • Jako Administrator lub planista • Chcę tworzyć i edytować wygenerowany plan lekcji na podstawie danych wejściowych takich jak przedmiot, ilość godzin w tygodniu i dostępny nauczyciel • bo wtedy/ponieważ otrzymam plan lekcji, który zawiera moja założenia. 		
Kryteria akceptacji	Interfejs pozwala na dodanie danych o przedmiocie i na ich podstawie wyświetla wygenerowany plan.		
Dane wejściowe	{uzupełniane w trakcie sprintu – dane wejściowe, związane z wymaganiem}		
Warunki początkowe	{ uzupełniane w trakcie sprintu – warunki, które muszą być prawdziwe przed wywołaniem operacji}		
Warunki końcowe	{ uzupełniane w trakcie sprintu – warunki, które muszą być prawdziwe po wywołaniu operacji}		
Sytuacje wyjątkowe	{ uzupełniane w trakcie sprintu – niepożądane sytuacje i sposoby ich obsługi}		
Szczegóły implementacji	{ uzupełniane w trakcie sprintu – opis sposobu realizacji}		
Udziałowiec	Administracja, Programiści		
Wymagania powiązane	F02, F05		

KARTA WYMAGANIA			
Identyfikator:	F02	Priorytet:	M
Nazwa	Moduł generujący plan		
Opis	<ul style="list-style-type: none"> • Jako Administrator lub planista • Chcę otrzymać wygenerowany plan • bo wtedy/ponieważ nie chcę poświęcać wielu godzin na ręczne stworzenie planu. 		
Kryteria akceptacji	Plan spełnia założenia ustanowione przez planistę, czyli przedmioty i liczba godzin w tygodniu dla danej grupy w wygenerowanym planie są zgodne z danymi, które wprowadzono.		
Dane wejściowe	{uzupełniane w trakcie sprintu – dane wejściowe, związane z wymaganiem}		
Warunki początkowe	{ uzupełniane w trakcie sprintu – warunki, które muszą być prawdziwe przed wywołaniem operacji}		
Warunki końcowe	{ uzupełniane w trakcie sprintu – warunki, które muszą być prawdziwe po wywołaniu operacji}		
Sytuacje wyjątkowe	{ uzupełniane w trakcie sprintu – niepożądane sytuacje i sposoby ich obsługi}		
Szczegóły implementacji	{ uzupełniane w trakcie sprintu – opis sposobu realizacji}		
Udziałowiec	Administracja, Programiści		
Wymagania powiązane	F01, F05		

KARTA WYMAGANIA			
Identyfikator:	F03	Priorytet:	M
Nazwa	Widok planu lekcji		
Opis	<ul style="list-style-type: none"> • Jako Administrator, nauczyciel lub student • Chcę widzieć plan lekcji • bo wtedy/ponieważ otrzymuje informację, na jakim przedmiocie, w jakiej sali i o jakiej godzinie powinienem być na zajęciach. 		
Kryteria akceptacji	Plan powinien wyświetlać dane na cały tydzień. Musi zawierać takie informacje, jak czas trwania zajęć, imię i nazwisko prowadzącego, czas i miejsce odbywających się zajęć.		
Dane wejściowe	{uzupełniane w trakcie sprintu – dane wejściowe, związane z wymaganiem}		
Warunki początkowe	{ uzupełniane w trakcie sprintu – warunki, które muszą być prawdziwe przed wywołaniem operacji}		
Warunki końcowe	{ uzupełniane w trakcie sprintu – warunki, które muszą być prawdziwe po wywołaniu operacji}		
Sytuacje wyjątkowe	{ uzupełniane w trakcie sprintu – niepożądane sytuacje i sposoby ich obsługi}		
Szczegóły implementacji	{ uzupełniane w trakcie sprintu – opis sposobu realizacji}		
Udziałowiec	Administracja, Programiści, nauczyciele, studenci		
Wymagania powiązane	F04, F06		

KARTA WYMAGANIA			
Identyfikator:	F04	Priorytet:	M
Nazwa	Widok zajęć nauczyciela		
Opis	<ul style="list-style-type: none"> • Jako nauczyciel • Chcę mieć w jednym miejscu plan, który informuje mnie jakie zajęcia mam w danej godzinie • bo wtedy/ponieważ mogę prowadzić zajęcia z planu. 		
Kryteria akceptacji	Nauczyciel otrzymuje plan swoich zajęć, zgodny z planami lekcji grup. Powinien być poinformowany o godzinie i sali, w której powinien być.		
Dane wejściowe	{uzupełniane w trakcie sprintu – dane wejściowe, związane z wymaganiem}		
Warunki początkowe	{ uzupełniane w trakcie sprintu – warunki, które muszą być prawdziwe przed wywołaniem operacji}		
Warunki końcowe	{ uzupełniane w trakcie sprintu – warunki, które muszą być prawdziwe po wywołaniu operacji}		
Sytuacje wyjątkowe	{ uzupełniane w trakcie sprintu – niepożądane sytuacje i sposoby ich obsługi}		
Szczegóły implementacji	{ uzupełniane w trakcie sprintu – opis sposobu realizacji}		
Udziałowiec	Nauczyciele, Programiści		
Wymagania powiązane	F03		

KARTA WYMAGANIA			
Identyfikator:	F05	Priorytet:	S
Nazwa	Ankieta czasu dostępności		
Opis	<ul style="list-style-type: none"> • Jako nauczyciel • Chcę na początku roku wprowadzić czas w tygodniu, kiedy mogę prowadzić zajęcia w szkole • bo wtedy/ponieważ mogę mieć plan, w którym są uwzględnione godziny dostępności. 		
Kryteria akceptacji	Nauczyciel nie powinien mieć zajęć w dniach i godzinach, kiedy nie jest dostępny.		
Dane wejściowe	{uzupełniane w trakcie sprintu – dane wejściowe, związane z wymaganiem}		
Warunki początkowe	{ uzupełniane w trakcie sprintu – warunki, które muszą być prawdziwe przed wywołaniem operacji}		
Warunki końcowe	{ uzupełniane w trakcie sprintu – warunki, które muszą być prawdziwe po wywołaniu operacji}		
Sytuacje wyjątkowe	{ uzupełniane w trakcie sprintu – niepożądane sytuacje i sposoby ich obsługi}		
Szczegóły implementacji	{ uzupełniane w trakcie sprintu – opis sposobu realizacji}		
Udziałowiec	Nauczyciele, Programiści		
Wymagania powiązane	F01,F02		

KARTA WYMAGANIA			
Identyfikator:	F06	Priorytet:	C
Nazwa	Tworzenie notatek		
Opis	<ul style="list-style-type: none"> • Jako nauczyciel lub administrator • Chcę jako nauczycielchcę powiadomić moich studentów o spóźnieniu lub nieobecności • bo wtedy/ponieważ potrzebuje w razie nagłych zmian poinformować o nich studentów. 		
Kryteria akceptacji	Nauczyciel może stworzyć notatkę, a studenci dostają powiadomienie oraz widzą notatkę na planie lekcji, którą mogą kliknąć, po czym dostaną treść wiadomości.		
Dane wejściowe	{uzupełniane w trakcie sprintu – dane wejściowe, związane z wymaganiem}		
Warunki początkowe	{ uzupełniane w trakcie sprintu – warunki, które muszą być prawdziwe przed wywołaniem operacji}		
Warunki końcowe	{ uzupełniane w trakcie sprintu – warunki, które muszą być prawdziwe po wywołaniu operacji}		
Sytuacje wyjątkowe	{ uzupełniane w trakcie sprintu – niepożądane sytuacje i sposoby ich obsługi}		
Szczegóły implementacji	{ uzupełniane w trakcie sprintu – opis sposobu realizacji}		
Udziałowiec	Nauczyciele, Programiści		
Wymagania powiązane	F04		

KARTA WYMAGANIA			
Identyfikator:	F07	Priorytet:	C
Nazwa	Panel informacyjny		
Opis	<ul style="list-style-type: none"> • Jako student • Chcę otrzymać informację o spóźnieniu lub nieobecności nauczyciela • bo wtedy/ponieważ potrzebuje wiedzieć, czy zajęcia się odbędą. 		
Kryteria akceptacji	Student otrzymuje powiadomienie push. Może kliknąć w notatkę wyświetlającą się na planie lekcji i przeczytać jej treść.		
Dane wejściowe	{uzupełniane w trakcie sprintu – dane wejściowe, związane z wymaganiem}		
Warunki początkowe	{ uzupełniane w trakcie sprintu – warunki, które muszą być prawdziwe przed wywołaniem operacji}		
Warunki końcowe	{ uzupełniane w trakcie sprintu – warunki, które muszą być prawdziwe po wywołaniu operacji}		
Sytuacje wyjątkowe	{ uzupełniane w trakcie sprintu – niepożądane sytuacje i sposoby ich obsługi}		
Szczegóły	{ uzupełniane w trakcie sprintu – opis sposobu realizacji}		

implementacji	
Udziałowiec	Studenci, Programiści
Wymagania powiązane	F04

3.2.2 Interfejs z otoczeniem

{Punkty styku projektowanego systemu z innymi podsystemami/sieciami/operatorami np. dopasowanie do baz danych, formatów, usług oferowanych przez inne, współpracujące systemy bądź serwisy; wymagania dot. dopasowania interfejsów itp.}

Dla EduPlaner:

KARTA WYMAGANIA			
Identyfikator:	WI01	Priorytet:	M – must (musi być)
Nazwa	System harmonogramowania zajęć		
Opis	System musi umożliwiać tworzenie harmonogramów zajęć dla szkół, uwzględniając dostępność sal, preferencje nauczycieli oraz ograniczenia czasowe uczniów. System powinien automatycznie wykrywać kolizje w harmonogramach i proponować alternatywne terminy.		
Kryteria akceptacji	Harmonogram generowany jest bez konfliktów czasowych; użytkownicy mogą przeglądać swoje indywidualne plany zajęć.		
Dane wejściowe	Informacje o dostępności sal, preferencje nauczycieli, czas wolny uczniów.		
Warunki początkowe	Wprowadzone dane o zasobach (sale, nauczyciele, klasy), preferencje czasowe.		
Warunki końcowe	Wygenerowany harmonogram bez konfliktów, dostępny do przeglądania przez nauczycieli, uczniów i dyрекcję.		
Sytuacje wyjątkowe	Brak dostępnych sal lub nauczycieli w danym terminie - system informuje administratora o konieczności ręcznej zmiany harmonogramu.		
Szczegóły implementacji	Algorytm optymalizacyjny uwzględniający preferencje i dostępność zasobów.		
Udziałowiec	Dyrekcja szkoły		
Wymagania powiązane	WI02 – system powiadomień dla nauczycieli i uczniów o zmianach w harmonogramie		

KARTA WYMAGANIA			
Identyfikator:	WI02	Priorytet:	S – should (powinno być)
Nazwa	System powiadomień		
Opis	System powinien wysyłać powiadomienia o aktualizacjach lub zmianach w harmonogramach bezpośrednio do użytkowników, takich jak nauczyciele i uczniowie. Powiadomienia mogą być wysyłane za pomocą aplikacji mobilnej lub wiadomości e-mail.		
Kryteria akceptacji	Użytkownicy są informowani o każdej zmianie w harmonogramie w ciągu 5 minut od jej zatwierdzenia.		
Dane wejściowe	Zmiany w harmonogramie zajęć.		
Warunki początkowe	Istniejące harmonogramy i zdefiniowane dane kontaktowe użytkowników.		
Warunki końcowe	Powiadomienia o zmianach wysłane do odpowiednich użytkowników.		
Sytuacje wyjątkowe	Brak dostępu do internetu – powiadomienia wysyłane ponownie przy przywróceniu łączności.		
Szczegóły implementacji	Moduł powiadomień integrujący się z e-mail oraz powiadomieniami push.		
Udziałowiec	Nauczyciele, uczniowie		
Wymagania powiązane	WI01 – system harmonogramowania zajęć		

KARTA WYMAGANIA			
Identyfikator:	WI03	Priorytet:	C – could (może być)
Nazwa	Eksport harmonogramu		
Opis	System może umożliwić eksport harmonogramu do formatów takich jak PDF czy CSV, aby ułatwić drukowanie i archiwizację.		
Kryteria akceptacji	Możliwość eksportu harmonogramu z poziomu aplikacji w formacie PDF lub CSV.		
Dane wejściowe	Harmonogram zajęć wygenerowany przez system.		
Warunki początkowe	Gotowy harmonogram dostępny w aplikacji.		
Warunki końcowe	Harmonogram w wybranym formacie pobrany przez użytkownika.		
Sytuacje wyjątkowe	Brak wystarczającej pamięci na urządzeniu użytkownika – informacja o konieczności zwolnienia miejsca.		
Szczegóły implementacji	Moduł eksportu danych do formatów PDF i CSV.		
Udziałowiec	Dyrekcja, nauczyciele		
Wymagania powiązane	WI01 – system harmonogramowania zajęć		

KARTA WYMAGANIA			
Identyfikator:	WI04	Priorytet:	W – won't (nie będzie – nie będzie implementowane w danym wydaniu, ale może być rozpatrzone w przyszłości)
Nazwa	Integracja z zewnętrznymi systemami edukacyjnymi		
Opis	W przyszłości system mógłby integrować się z innymi systemami edukacyjnymi, takimi jak e-dzienniki, co pozwoliłoby na automatyczną synchronizację danych o uczniach, nauczycielach i planach zajęć. Obecnie jednak brak takich integracji.		
Kryteria akceptacji	W przyszłości system umożliwi integrację z co najmniej jednym systemem e-dziennika poprzez API.		
Dane wejściowe	Dane z zewnętrznych systemów edukacyjnych (np. listy uczniów, oceny, obecności).		
Warunki początkowe	Dostęp do API systemów zewnętrznych oraz odpowiednie uprawnienia dostępu do danych.		
Warunki końcowe	Synchronizacja danych między EduPlanerem a systemem zewnętrznym (po wdrożeniu w przyszłości).		
Sytuacje wyjątkowe	Brak dostępu do API – informacja dla administratora o problemach z integracją.		
Szczegóły implementacji	Implementacja dodatkowego modułu integracji API (do rozważenia w przyszłości).		
Udziałowiec	Dyrekcja szkoły, nauczyciele		
Wymagania powiązane	WI01 – system harmonogramowania zajęć		

KARTA WYMAGANIA			
Identyfikator:	WI05	Priorytet:	M
Nazwa	Integracja widoku strony internetowej z backendem		
Opis	Graficzny Interfejs użytkownika w formie aplikacji WEB integruje się z backendową architekturą.		
Kryteria akceptacji	Istnieje komunikacja pomiędzy frontendem, a backendem.		
Dane wejściowe	{uzupełniane w trakcie sprintu – dane wejściowe, związane z wymaganiem}		
Warunki początkowe	{uzupełniane w trakcie sprintu – dane wejściowe, związane z wymaganiem}		
Warunki końcowe	{uzupełniane w trakcie sprintu – dane wejściowe, związane z wymaganiem}		
Sytuacje wyjątkowe	{uzupełniane w trakcie sprintu – dane wejściowe, związane z wymaganiem}		

Szczegóły implementacji	{uzupełniane w trakcie sprintu – dane wejściowe, związane z wymaganiem}
Udziałowiec	Dyrekcja szkoły, nauczyciele, studenci
Wymagania powiązane	WI01 – system harmonogramowania zajęć

KARTA WYMAGANIA			
Identyfikator:	WI06	Priorytet:	M – <i>must</i> (musi być)
Nazwa	Połączenie backend z bazą danych na serwerze		
Opis	System musi zapewniać stabilne połączenie między backendem a bazą danych na serwerze. Połączenie powinno wspierać jednocześnie zapytania od wielu użytkowników oraz być zgodne z wymaganiami dotyczącymi integralności i ochrony danych.		
Kryteria akceptacji	Połączenie z bazą danych na serwerze działa poprawnie i jest dostępne dla backendu. Transakcje CRUD (Create, Read, Update, Delete) są obsługiwane zgodnie z założeniami funkcjonalnymi systemu. Połączenie jest bezpieczne i odporne na nieautoryzowany dostęp oraz zapewnia szyfrowanie danych podczas transmisji.		
Dane wejściowe	{uzupełniane w trakcie sprintu – dane wejściowe, związane z wymaganiem}		
Warunki początkowe	{ uzupełniane w trakcie sprintu – warunki, które muszą być prawdziwe przed wywołaniem operacji}		
Warunki końcowe	{ uzupełniane w trakcie sprintu – warunki, które muszą być prawdziwe po wywołaniu operacji}		
Sytuacje wyjątkowe	{ uzupełniane w trakcie sprintu – niepożądane sytuacje i sposoby ich obsługi}		
Szczegóły implementacji	{ uzupełniane w trakcie sprintu – opis sposobu realizacji}		
Udziałowiec	Zespół programistyczny		
Wymagania powiązane	WI02 - Integracja systemu z bazą danych WI03 - Bezpieczeństwo połączeń i danych w systemie		

3.3 Wymagania pozafunkcjonalne

{Wymagania pozafunkcjonalne odpowiadają na pytanie: jak system ma działać? Obejmują one wymagania jakościowe: niezawodność, wiarygodność, wydajność, kategorie ergonomiczne: łatwość użycia, intuicyjność, modyfikowalność, adaptowalność itp. Wymagania te powinny być wyrażone w mierzalnej formie. Przykładowo, zamiast: „Wysoka dostępność systemu”, powinno być: „System powinien być dostępny 7 dni w tygodniu, 24 godziny na dobę”. zamiast: „System ma być łatwy w użyciu”, powinno być: „System musi umożliwiać sprawne wprowadzanie dużych ilości masowych danych” albo: „System ma być intuicyjny do użycia przez niedoświadczonego użytkownika. Powinien być zaopatrzony w instrukcję, pliki pomocy i objaśnienia pozycji menu i klawiszy funkcyjnych.”}

KARTA WYMAGANIA			
Identyfikator:	NF01	Priorytet:	M
Nazwa	Niezwadność		
Opis	System musi być dostępny 24/7		
Kryteria akceptacji	Warunki Satysfakcji: testujemy kilkakrotnie o różnych porach, z różnych lokalizacji		
Udziałowiec	Programiści		
Wymagania powiązane	-		

KARTA WYMAGANIA			
Identyfikator:	NF02	Priorytet:	S
Nazwa	Intuicyjność dla nowych użytkowników		

Opis	Nowy użytkownik powinien być w stanie wykonać podstawowe operacje (np. przeglądanie harmonogramu, zgłaszanie preferencji) bez dodatkowego przeszkolenia.
Kryteria akceptacji	Warunki Satysfakcji: musimy dostać pozytywną opinie użytkowników testujących.
Udziałowiec	Programiści
Wymagania powiązane	-

KARTA WYMAGANIA			
Identyfikator:	NF03	Priorytet:	M
Nazwa	Kompatybilność przeglądarek		
Opis	System powinien być kompatybilny z najnowszymi wersjami popularnych przeglądarek (Chrome, Firefox, Safari).		
Kryteria akceptacji	Warunki Satysfakcji: testujemy kilkakrotnie na Chrome, Firefox i Safari		
Udziałowiec	Programiści		
Wymagania powiązane	-		

KARTA WYMAGANIA			
Identyfikator:	NF04	Priorytet:	C
Nazwa	Uwierzytelnianie i autoryzacja		
Opis	System musi zapewniać logowanie z podziałem na role (dyrekcja, nauczyciele, uczniowie)		
Kryteria akceptacji	Warunki Satysfakcji: Testy End2End mają zakończyć się sukcesem.		
Udziałowiec	Programiści		
Wymagania powiązane	-		

KARTA WYMAGANIA			
Identyfikator:	NF05	Priorytet:	C
Nazwa	Możliwość konfiguracji harmonogramów		
Opis	System musi pozwalać administratorom na modyfikację parametrów harmonogramu (np. preferencje nauczycieli, czas zajęć) bez konieczności modyfikacji kodu, co umożliwia szybkie dostosowanie do zmieniających się wymagań.		
Kryteria akceptacji	Warunki Satysfakcji: Testy End2End mają zakończyć się sukcesem.		
Udziałowiec	Programiści		
Wymagania powiązane	-		

KARTA WYMAGANIA			
Identyfikator:	NF06	Priorytet:	C
Nazwa	Prosty interfejs dla administratorów:		
Opis	Administratorzy powinni mieć dostęp do wszystkich kluczowych funkcji jak tworzenie kont i zarządzanie uprawnieniami		
Kryteria akceptacji	Warunki Satysfakcji: testujemy End2End		
Udziałowiec	Administracja		
Wymagania powiązane	-		

3.4 Wymagania na środowisko docelowe

{W jakim środowisku będzie pracować system – o ile jest istotne, np. system operacyjny, rodzaje i wersje przeglądarek internetowych, itp. Może się zdarzyć, że na tym etapie użytkownicy i inni udziałowcy nie wyspecyfikują środowiska docelowego.}

System EduPlaner będzie działać w środowisku webowym, co pozwoli na dostęp z różnych urządzeń i systemów operacyjnych. Aplikacja powinna być kompatybilna z następującymi środowiskami:

KARTA WYMAGANIA			
Identyfikator:	ŚD01	Priorytet:	M – must (musi być)
Nazwa	Wymagania środowiskowe systemu		
Opis	System EduPlaner ma działać w środowisku webowym, umożliwiając dostęp z urządzeń mobilnych i komputerów stacjonarnych. System musi być kompatybilny z najpopularniejszymi systemami operacyjnymi i przeglądarkami internetowymi oraz działać w oparciu o minimalne wymagania sprzętowe i sieciowe, zapewniając płynne i stabilne użytkowanie.		
Kryteria akceptacji	System poprawnie działa na systemach operacyjnych Windows 10+, MacOS 11+, Android 8.0+, iOS 14+. Aplikacja działa na przeglądarkach: Google Chrome (90+), Mozilla Firefox (85+), Microsoft Edge (88+), Safari (14+). Przy minimalnej przepustowości 1 Mbps aplikacja ładuje się w czasie krótszym niż 3 sekundy.		
Udziałowiec	Zespół projektowy EduPlaner		
Wymagania powiązane	Brak		

4 Odwołania do literatury

{Lista przywoływanych pozycji literowych, ponumerowanych lub z przydzielonymi identyfikatorami; w treści właściwej dokumentu posługujemy się wyłącznie numerami/ identyfikatorami do wskazania źródła treści. Usunąć jeśli nie dotyczy.}

Nie dotyczy

5 Załączniki