

Title

Polsko-Japońska Akademia Technik Komputerowych

University Calendar

czyli system wspomagający w zarządzaniu zajęciami, grupami oraz studentami uczelni

Author:

Zahar Zubyk s24984

Sporządzono:

czerwiec 2024

Spis treści

Title

Polsko-Japońska Akademia Technik Komputerowych

University Calendar

Author:

Sporządzono:

1. Wymagania użytkownika

Dziedzina biznesowa i cel

Zakres odpowiedzialności systemu

Użytkownicy systemu

Wymagania użytkownika

1. Język Aplikacji
2. Dostępność
3. Logowanie
4. Zarządzanie Grupami, Studentami i Zajęciami
5. Przeglądanie Planu Zajęć
6. Sortowanie Zajęć
7. Panel Administracyjny

Interfejs Użytkownika

1. Ekran Logowania
2. Panel Administratora
3. Plan Zajęć

Wymagania niefunkcjonalne

Słownik pojęć

Podsumowanie

Wymagania techniczne

1. Środowisko działania
2. Baza danych
3. Bezpieczeństwo

Funkcjonalności

1. Zorganizowanie zajęć przez dyrektora
2. Zmiana statusu zajęć
3. Rejestracja wyników
4. Zarejestrowanie studenta w grupie
5. Usunięcie studenta z grupy
6. Edycja danych

Ograniczenia i inwarianty

1. Role użytkowników
2. Walidacja danych
3. Synchronizacja danych

Konkretnie przechowywane dane

1. Osoba
2. Student
3. Dydaktyk
4. Dyrektor
5. Zajęcia
6. Grupa

Podsumowanie

2. Diagram przypadków użycia

3. Diagram klas – analityczny

4. Diagram klas - projektowy

5. Scenariusz przypadków użycia

Przypadek Użycia 1: Logowanie do Systemu

Przypadek Użycia 2: Dodawanie Grupy

Przypadek Użycia 3: Usuwanie Studenta z Grupy

Przypadek Użycia 4: Report student

6. Diagram aktywności dla przypadków użycia

Dyrektor wyrzuca grupę z systemu

7. Diagramy stanu dla klas

Logowanie

Grupa

Student

8. Projekt GUI

Strona logowania

Strona Studenta / Dydaktyka

Strona Dyrektora

9. Omówienie decyzji projektowych i skutków analizy dynamicznej

1. Wymagania użytkownika

Dziedzina biznesowa i cel

Projekt ma na celu stworzenie systemu do zarządzania planem zajęć dla studentów, dydaktyków oraz administracji Uniwersytetu PJATK. Obecnie używane narzędzia są przestarzałe i nie spełniają wymagań użytkowników. Nowy system ma być intuicyjny, łatwy w obsłudze oraz posiadać zaawansowane funkcjonalności, które ułatwią zarządzanie zajęciami, grupami i studentami.

Zakres odpowiedzialności systemu

System ma umożliwiać użytkownikom (studentom, dydaktykom, dyrektorowi) sprawne tworzenie i zarządzanie planami zajęć, przechowywanie niezbędnych informacji, rejestrowanie i monitorowanie działań. System ma również umożliwiać edycję danych w celu wprowadzania poprawek oraz przechowywać dane w trwałej formie, aby można je było odtworzyć przy kolejnym uruchomieniu programu.

Użytkownicy systemu

System powinien być w stanie przełączać się między widokami dla studentów, dydaktyków i dyrektora. Widoki te mają być wspólne, ale różnić się dostępnymi funkcjami. Wszyscy użytkownicy będą korzystać z tej samej aplikacji na tej samej maszynie, ale z różnymi uprawnieniami.

Wymagania użytkownika

1. Język Aplikacji

- System musi być dostępny w języku angielskim.

2. Dostępność

- System dostępny dla studentów, dydaktyków oraz dyrektora.

3. Logowanie

- Każdy użytkownik (student, dydaktyk, dyrektor) musi mieć unikalny indeks i hasło, które będą wymagane do zalogowania się do systemu.

4. Zarządzanie Grupami, Studentami i Zajęciami

- Dyrektor musi mieć możliwość dodawania i usuwania grup.
- Dyrektor musi mieć możliwość dodawania i usuwania studentów z grup.
- System powinien przechowywać informacje o prowadzącym, unikalnym ID zajęcia, czasie rozpoczęcia i zakończenia, temacie zajęcia oraz informacji, czy to ćwiczenia czy wykład.
- Zajęcia ćwiczeniowe muszą zawierać informacje o sali i przedmiocie, a wykłady o auli i przedmiocie.

5. Przeglądanie Planu Zajęć

- Studenci i dydaktycy muszą mieć możliwość przeglądania planu zajęć, w tym zajęć innych grup.
- Użytkownicy muszą mieć możliwość przeglądania planu zajęć w formatach: dzień, tydzień, miesiąc.
- System musi zawierać informacje o dniach wolnych.

6. Sortowanie Zajęć

- Studenci i dydaktycy muszą mieć możliwość sortowania zajęć według prowadzących lub grup.

7. Panel Administracyjny

- Panel administracyjny musi umożliwiać dyrektorowi dodawanie i usuwanie grup, studentów oraz zajęć.

Interfejs Użytkownika

1. Ekran Logowania

- Każdy użytkownik (student, dydaktyk, dyrektor) ma unikalny indeks i hasło, które jest wymagane dla logowania.

2. Panel Administratora

- Zarządzanie grupami, studentami oraz zajęciami.

3. Plan Zajęć

- Użytkownicy mogą przeglądać plan zajęć w formatach: dzień, tydzień, miesiąc.

Wymagania niefunkcjonalne

1. Łatwość użytkowania

- System powinien być prosty i intuicyjny w użyciu. Funkcje systemu należy odpowiednio pogrupować i opisać, aby niedoświadczony w użytkowaniu komputera klient mógł z łatwością opanować funkcje systemu. Kluczowe funkcjonalności powinny być logicznie rozmieszczone i dostępne w podobny sposób jak w innych popularnych systemach zarządzania, aby użytkownicy mogli się łatwo przestawić na nasz system. Czas szkolenia użytkownika nie powinien przekraczać godziny. Należy stworzyć okienka z podpowiedziami dla funkcjonalności, które mogą być mniej intuicyjne.

2. Niezawodność

- System powinien działać niezawodnie przez co najmniej 24 miesiące od uruchomienia, zakładając wystarczającą ilość miejsca w pamięci komputera, na którym system jest zainstalowany. W przypadku awarii nie z winy użytkownika, naprawa systemu nie powinna zająć dłużej niż jeden dzień. System musi być odporny na typowe błędy użytkownika i minimalizować ryzyko wystąpienia krytycznych awarii.

3. Bezpieczeństwo danych

- System powinien przechowywać dane w sposób bezpieczny, uniemożliwiający ich wyciek zgodnie z uznanymi standardami bezpieczeństwa na dzień sporządzania dokumentu. Dane użytkowników, szczególnie informacje logowania i dane osobowe, powinny być szyfrowane. Wstępna implementacja bazy danych będzie oparta na MySQL w najnowszej wersji, ale zakłada się możliwość zmiany silnika bazy danych w przyszłości.

4. Łatwa modyfikowalność

- System powinien wspierać łatwą modyfikowalność, aby poprawnie odwzorowywać dynamicznie zmieniające się zasady i wymagania użytkowników. Każda nowa wersja systemu powinna umożliwiać łatwe dodawanie i usuwanie funkcjonalności bez konieczności przebudowy całego systemu. Dokumentacja kodu powinna być czytelna i zrozumiała, aby umożliwić przyszłym programistom szybkie wprowadzenie niezbędnych zmian.

5. Wydajność

- System powinien działać płynnie i szybko, nawet przy dużej liczbie jednoczesnych użytkowników. Czas ładowania stron nie powinien przekraczać 3 sekund. System musi być skalowalny, aby w przyszłości mógł obsłużyć większą liczbę użytkowników i danych bez utraty wydajności.

6. Dostępność

- System powinien być dostępny 24/7 z minimalnym czasem przestojów, nie przekraczającym 2 godzin miesięcznie na konserwację. Należy zadbać o regularne kopie zapasowe danych, aby w razie awarii można było szybko przywrócić pełną funkcjonalność systemu.

7. Interoperacyjność

- System powinien być kompatybilny z innymi systemami używanymi w uniwersytecie, takimi jak systemy rejestracji studentów, systemy płatności i inne.

Słownik pojęć

1. **Student:** osoba, która jest zarejestrowana w systemie jako uczestnik zajęć.
2. **Dydaktyk:** osoba prowadząca zajęcia dla studentów.
3. **Dyrektor:** osoba zarządzająca planem zajęć i mająca dostęp do panelu administracyjnego.
4. **Grupa:** zbiór studentów przypisanych do określonych zajęć.
5. **Zajęcia:** spotkanie edukacyjne, które może być wykładem lub ćwiczeniami, z określonym prowadzącym, czasem trwania i miejscem.
6. **Plan zajęć:** harmonogram zajęć dla studentów i dydaktyków, dostępny do przeglądania w różnych formatach czasowych.
7. **Interfejs użytkownika:** sposób, w jaki użytkownik wchodzi w interakcję z systemem, w tym ekrany logowania, panel administracyjny i widok planu zajęć.

Podsumowanie

Zaprojektowany system zapewni efektywne zarządzanie planem zajęć Uniwersytetu PJATK, uwzględniając potrzeby studentów, dydaktyków oraz administracji. Dzięki przejrzystemu interfejsowi użytkownicy będą mogli łatwo przeglądać i zarządzać swoimi zajęciami, co wpłynie na zwiększenie efektywności procesu nauczania i organizacji zajęć.

Wymagania techniczne

1. Środowisko działania

- System powinien być dostępny jako aplikacja webowa, działająca na wszystkich popularnych przeglądarkach internetowych.

2. Baza danych

- System powinien korzystać z relacyjnej bazy danych do przechowywania informacji o użytkownikach, grupach, zajęciach i innych istotnych danych.

3. Bezpieczeństwo

- System musi zapewniać bezpieczne logowanie oraz ochronę danych osobowych zgodnie z obowiązującymi przepisami.

Funkcjonalności

1. Zorganizowanie zajęć przez dyrektora

- Możliwość dodawania i usuwania zajęć z planu.

2. Zmiana statusu zajęć

- Możliwość zmiany statusu zajęć (zaplanowane, w trakcie, zakończone).

3. Rejestracja wyników

- Możliwość wprowadzania wyników zajęć przez dydaktyków.

4. Zarejestrowanie studenta w grupie

- Możliwość przypisania studenta do grupy.

5. Usunięcie studenta z grupy

- Możliwość usunięcia studenta z grupy.

6. Edycja danych

- Możliwość edycji informacji o zajęciach, studentach, grupach i innych elementach.

Ograniczenia i inwarianty

1. Role użytkowników

- Każdy użytkownik może mieć tylko jedną z ról (student, dydaktyk, dyrektor) w kontekście tego samego zajęcia.

2. Walidacja danych

- System powinien sprawdzać poprawność wprowadzanych danych i informować o błędach.

3. Synchronizacja danych

- System powinien zapewniać synchronizację danych pomiędzy różnymi widokami i użytkownikami w czasie rzeczywistym.

Konkretnie przechowywane dane

1. Osoba

- Imię, nazwisko, przynależność do organizacji (opcjonalnie).

2. Student

- Numer indeksu, zdobyte punkty (wliczalne na podstawie wyników z zajęć).

3. Dydaktyk

- Numer pracownika, stopień naukowy.

4. Dyrektor

- Numer pracownika, zakres odpowiedzialności.

5. Zajęcia

- ID zajęcia, prowadzący, czas rozpoczęcia i zakończenia, temat, rodzaj (ćwiczenia/wykład).

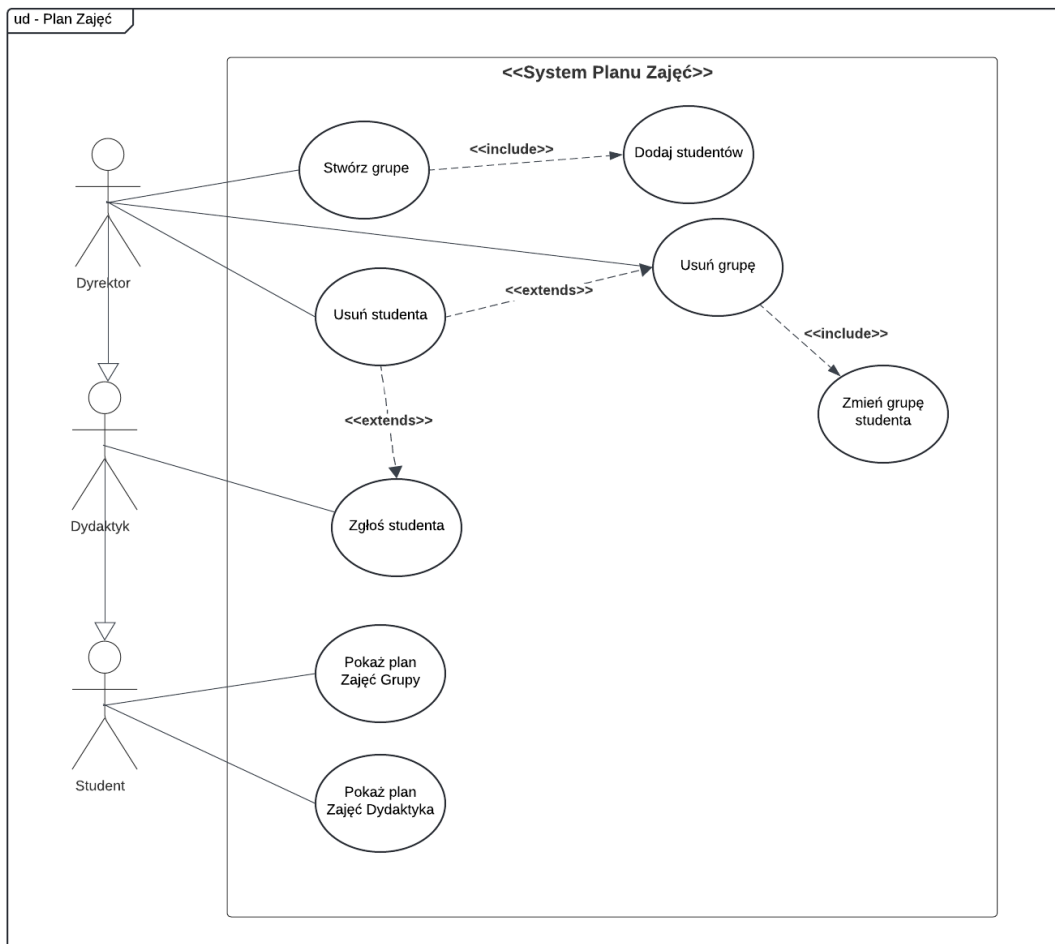
6. Grupa

- ID grupy, lista studentów.

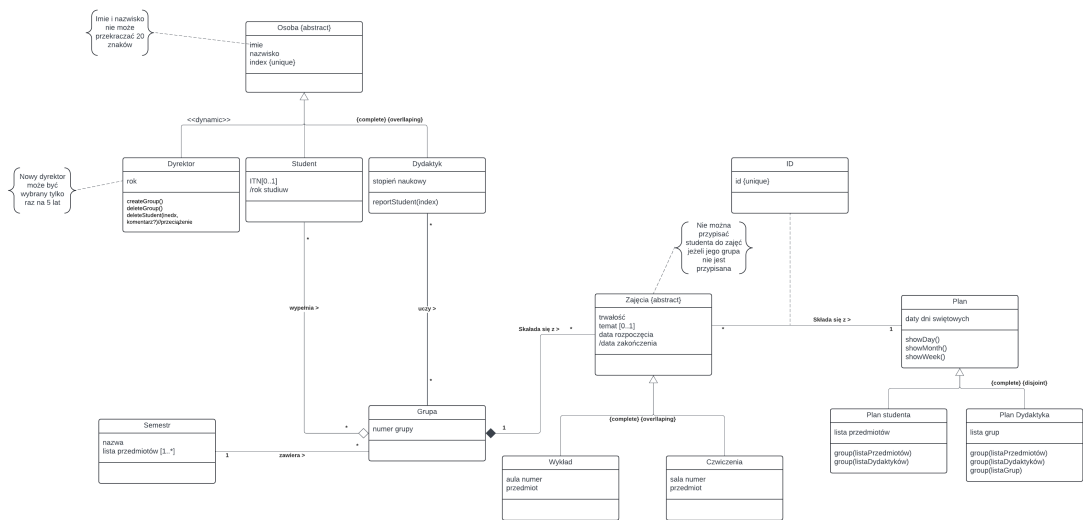
Podsumowanie

Nowy system zarządzania planem zajęć dla Uniwersytetu PJATK ma na celu poprawę efektywności organizacji i zarządzania zajęciami, grupami oraz studentami. Dzięki nowoczesnym funkcjonalnościom i intuicyjnemu interfejsowi, system będzie wspierał pracę dydaktyków, studentów oraz administracji, przyczyniając się do lepszej organizacji procesu nauczania.

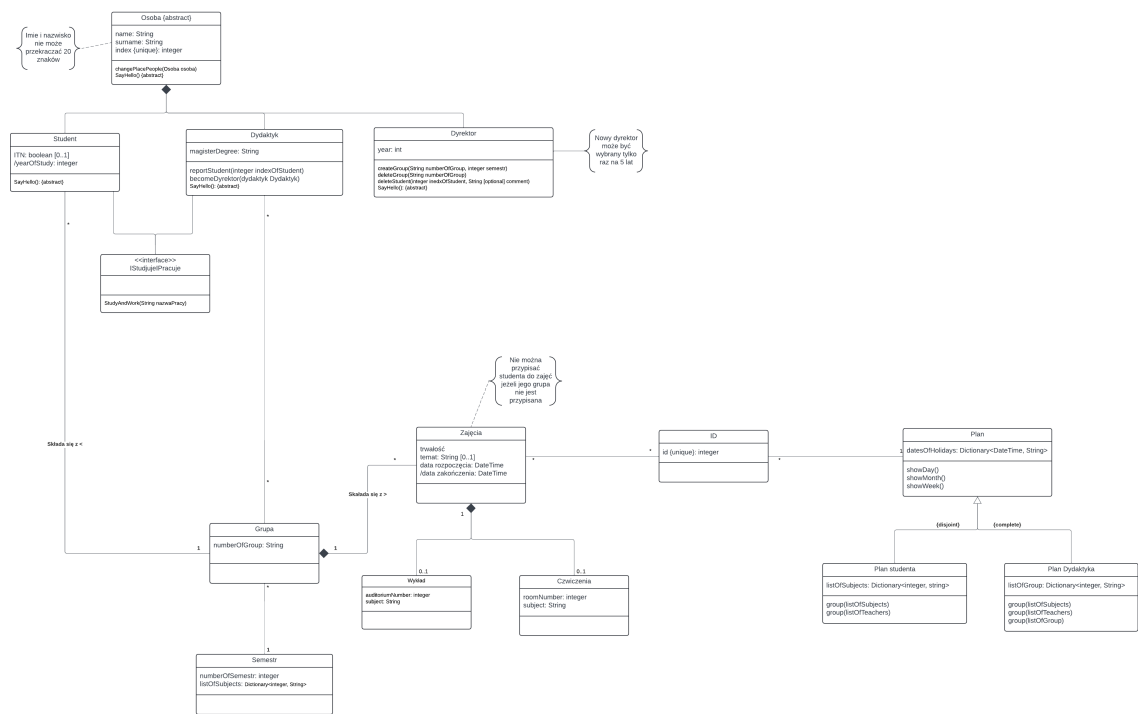
2. Diagram przypadków użycia



3. Diagram klas – analityczny



4. Diagram klas - projektowy



5. Scenariusz przypadków użycia

Przypadek Użycia 1: Logowanie do Systemu

Aktorzy: Student, Dydaktyk, Dyrektor

Opis: Użytkownik loguje się do systemu, aby uzyskać dostęp do swoich funkcji.

Przebieg główny:

1. Użytkownik otwiera ekran logowania.
2. Użytkownik wprowadza swój unikalny indeks i hasło.
3. System weryfikuje dane logowania.
4. System przekierowuje użytkownika do odpowiedniego panelu (panel studenta, dydaktyka lub dyrektora).

Alternatywny przebieg:

- 3a. Jeśli dane logowania są niepoprawne, system wyświetla komunikat o błędzie i prosi o ponowne wprowadzenie danych.

Przypadek Użycia 2: Dodawanie Grupy

Aktorzy: Dyrektor

Opis: Dyrektor dodaje nową grupę do systemu.

Przebieg główny:

1. Dyrektor loguje się do panelu administracyjnego.
2. Dyrektor wybiera opcję „Dodaj grupę”.
3. Dyrektor wprowadza dane nowej grupy (nazwa, opis, itp.).
4. Dyrektor zatwierdza dodanie grupy.
5. System zapisuje nową grupę i potwierdza operację.

Alternatywny przebieg:

- 3a. Jeśli dane grupy są już zapisane w bazie danych
- 4a. Jeśli dane grupy są niekompletne, system wyświetla komunikat o błędzie i prosi o uzupełnienie brakujących informacji.

Przypadek Użycia 3: Usuwanie Studenta z Grupy

Aktorzy: Dyrektor

Opis: Dyrektor usuwa studenta z istniejącej grupy.

Przebieg główny:

1. Dyrektor loguje się do panelu administracyjnego.

2. Dyrektor wybiera grupę, z której chce usunąć studenta.
3. Dyrektor wybiera studenta do usunięcia.
4. Dyrektor zatwierdza usunięcie studenta.
5. System usuwa studenta z grupy i potwierdza operację.

Alternatywny przebieg:

- 4a. Jeśli student nie istnieje w wybranej grupie, system wyświetla komunikat o błędzie.

Przypadek Użycia 4: Report student

Aktorzy: Dydaktyk

Opis: Użytkownik przegląda listę studentów i może zgłosić studenta.

Przebieg główny:

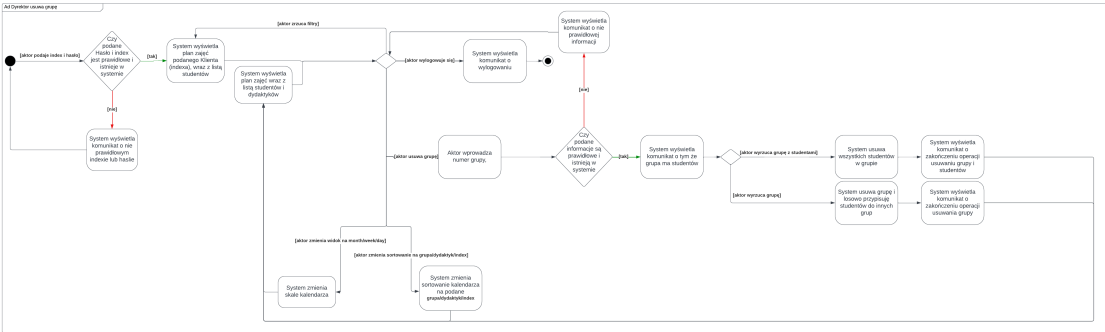
1. Użytkownik loguje się do systemu.
2. Użytkownik wybiera opcję „Przeglądaj listę studentów”.
3. Użytkownik wybiera sortowanie (grupy)
4. System wyświetla komunikat o zgłoszeniu studenta.

Alternatywny przebieg:

- -

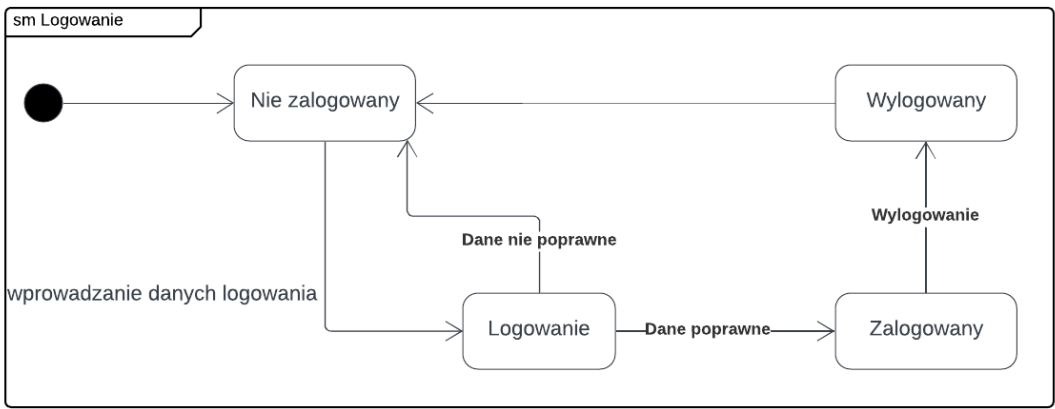
6. Diagram aktywności dla przypadków użycia

Dyrektor wyrzuca grupę z systemu

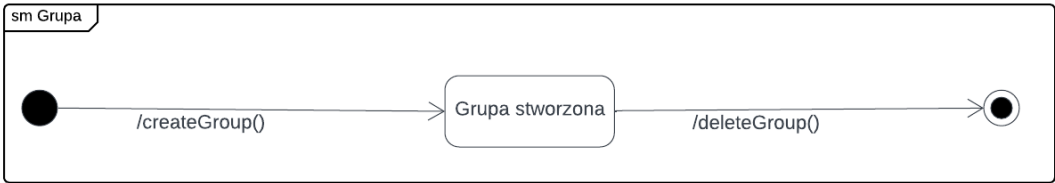


7. Diagramy stanu dla klas

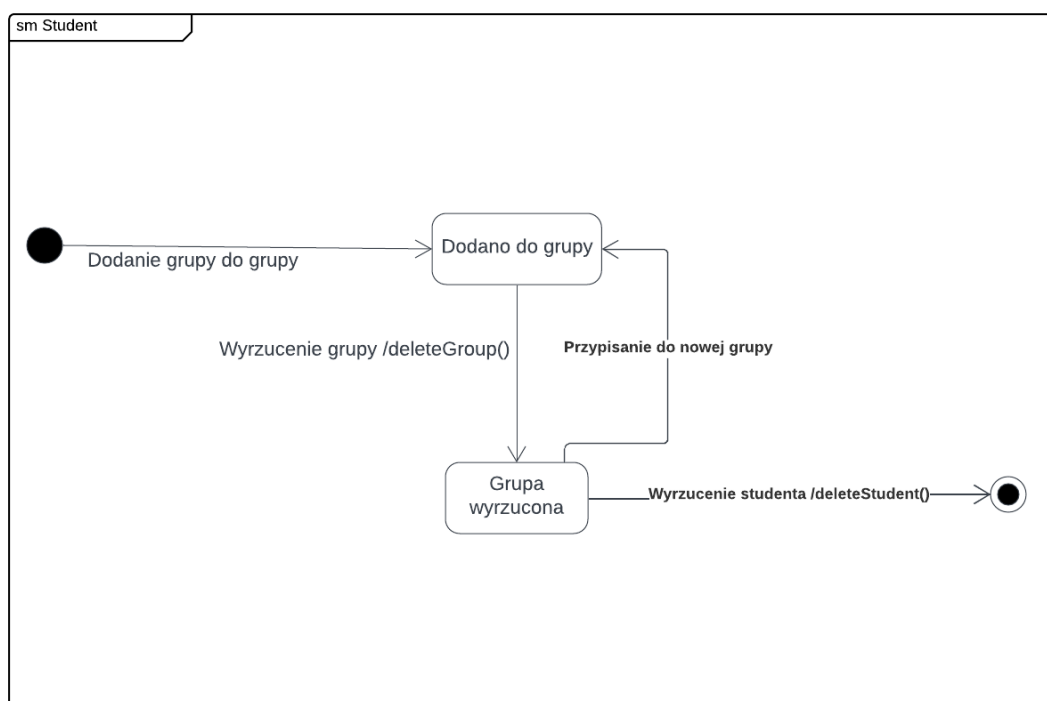
Logowanie



Grupa



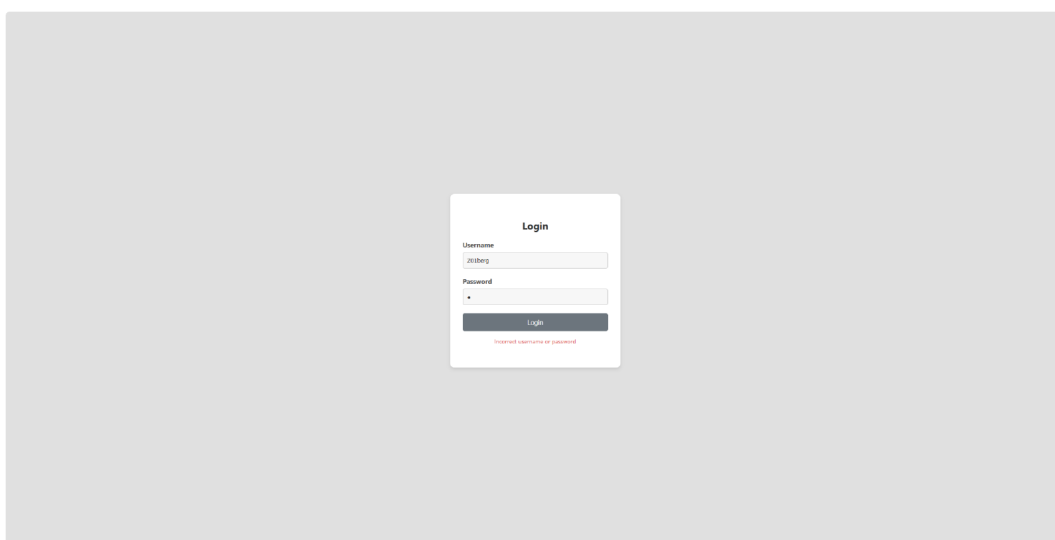
Student



8. Projekt GUI

Strona logowania

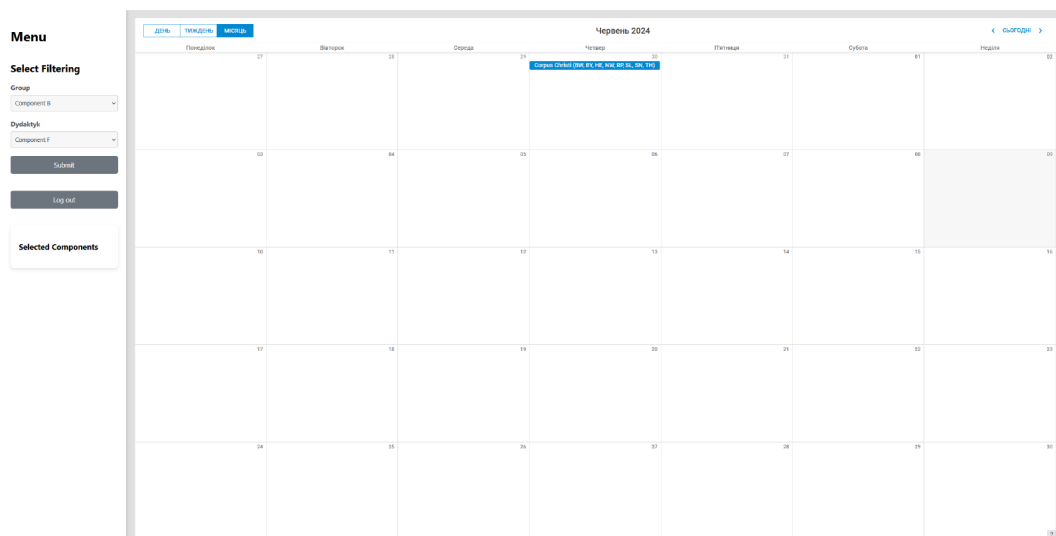
Końcowy wygląd może ulec zmianie



Jest to pierwszy widok, który pojawia się po wejściu na stronę www. Z tego poziomu widzimy dwa pola i przycisk. Username, Password i przycisk Login.

Strona Studenta / Dydaktyka

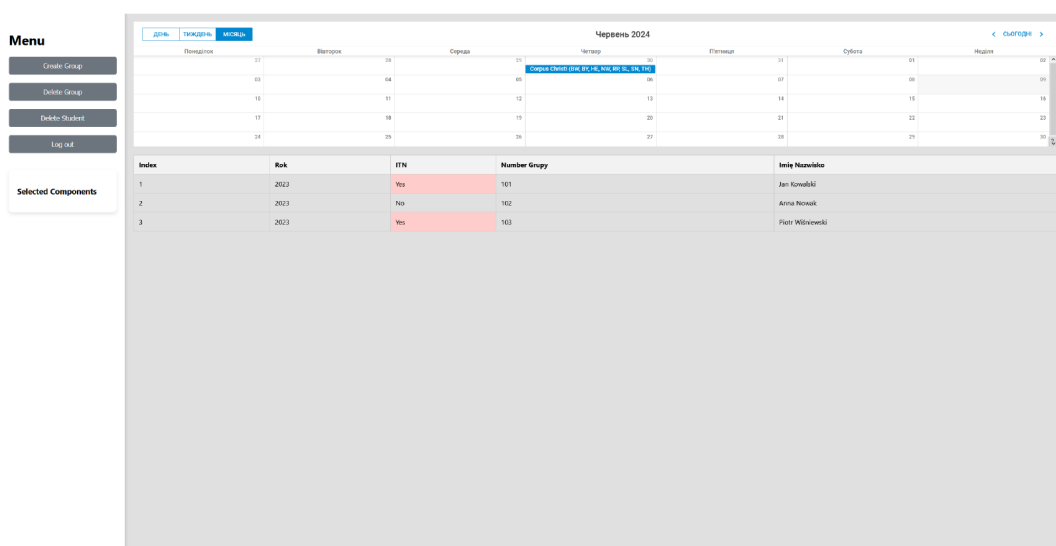
Końcowy wygląd może ulec zmianie



Po zalogowaniu do systemu jako Dydaktyk / Student mamy widok na kalendarz z polami do sortowania i przyciskiem do wylogowania

Strona Dyrektora

Końcowy wygląd może ulec zmianie



Po zalogowaniu do systemu jako Dyrektor mamy widok na kalendarz z polami do sortowania i przyciskiem do wylogowania. Dodatkowo też mamy dostęp do 3 nowych przycisków Stwórz grupę, Wyrzuć grupę, Wyrzuć studenta. Pod planem zajęć też mamy listę studentów lub listę grup w zależności od wybranej opcji.

9. Omówienie decyzji projektowych i skutków analizy dynamicznej

W projekcie systemu zarządzania zajęciami, grupami oraz studentami uczelni, podjęto kilka kluczowych decyzji projektowych, które mają wpływ na jego funkcjonalność, wydajność, skalowalność i bezpieczeństwo. Poniżej przedstawiono główne decyzje projektowe oraz ich uzasadnienie:

1. Wybór technologii front-endowej

- **Decyzja:** Zastosowanie technologii HTML/CSS/JS do budowy interfejsu użytkownika.
- **Uzasadnienie:** HTML, CSS i JavaScript to podstawowe technologie webowe, które są szeroko stosowane i dobrze znane. Pozwalają na tworzenie responsywnych i przyjaznych dla użytkownika interfejsów, które są kompatybilne z różnymi przeglądarkami i urządzeniami. Dodatkowo, duża liczba dostępnych bibliotek i frameworków, takich jak Bootstrap czy jQuery, umożliwia szybki rozwój i łatwe dostosowywanie interfejsu.

2. Wybór technologii back-endowej

- **Decyzja:** Użycie ASP.NET Core Web App (Model-View-Controller) do budowy aplikacji back-endowej.
- **Uzasadnienie:** ASP.NET Core to nowoczesny, wysokowydajny framework do tworzenia aplikacji webowych. Model-View-Controller (MVC) umożliwia wyraźne oddzielenie logiki aplikacji od interfejsu użytkownika, co zwiększa przejrzystość kodu i ułatwia jego utrzymanie. ASP.NET Core jest również dobrze zintegrowany z różnymi narzędziami Microsoftu, co ułatwia wdrażanie i zarządzanie aplikacją.

3. Wybór bazy danych

- **Decyzja:** Wykorzystanie MySQL jako silnika bazy danych.
- **Uzasadnienie:** MySQL jest jednym z najpopularniejszych i najbardziej niezawodnych systemów zarządzania relacyjnymi bazami danych. Oferuje wysoką wydajność, skalowalność oraz bezpieczeństwo danych. Jego szeroka akceptacja w branży oraz dostępność wsparcia technicznego dodatkowo przemawiają za tym wyborem.

4. Zastosowanie frameworka OWC

- **Decyzja:** Wykorzystanie Open Web Calendar.
- **Uzasadnienie:** OWC (Open Web Calendar) ułatwia zarządzanie danymi kalendarza, co jest kluczowe w systemie zarządzania zajęciami. Framework ten oferuje funkcje interaktywnego zarządzania wydarzeniami, synchronizację z różnymi kalendarzami oraz możliwość integracji z innymi systemami, co zwiększa produktywność i użyteczność systemu.