

Podstawy Systemów Bazodanowych - projekt

System zapisów na WF

Popowski Andrzej, Szatkowski Filip, Zawrotny Maksym
gr 33ip

27 czerwca 2018

Członek zespołu	Rola
Popowski Andrzej	model relacyjny bazy danych, projekt ekranów, diagramy czynności, określenie raportów
Szatkowski Filip	implementacja bazy danych wraz z interfejsem użytkownika
Zawrotny Maksym	notatka, model EER, aktorzy, przypadki użycia, dokumentacja

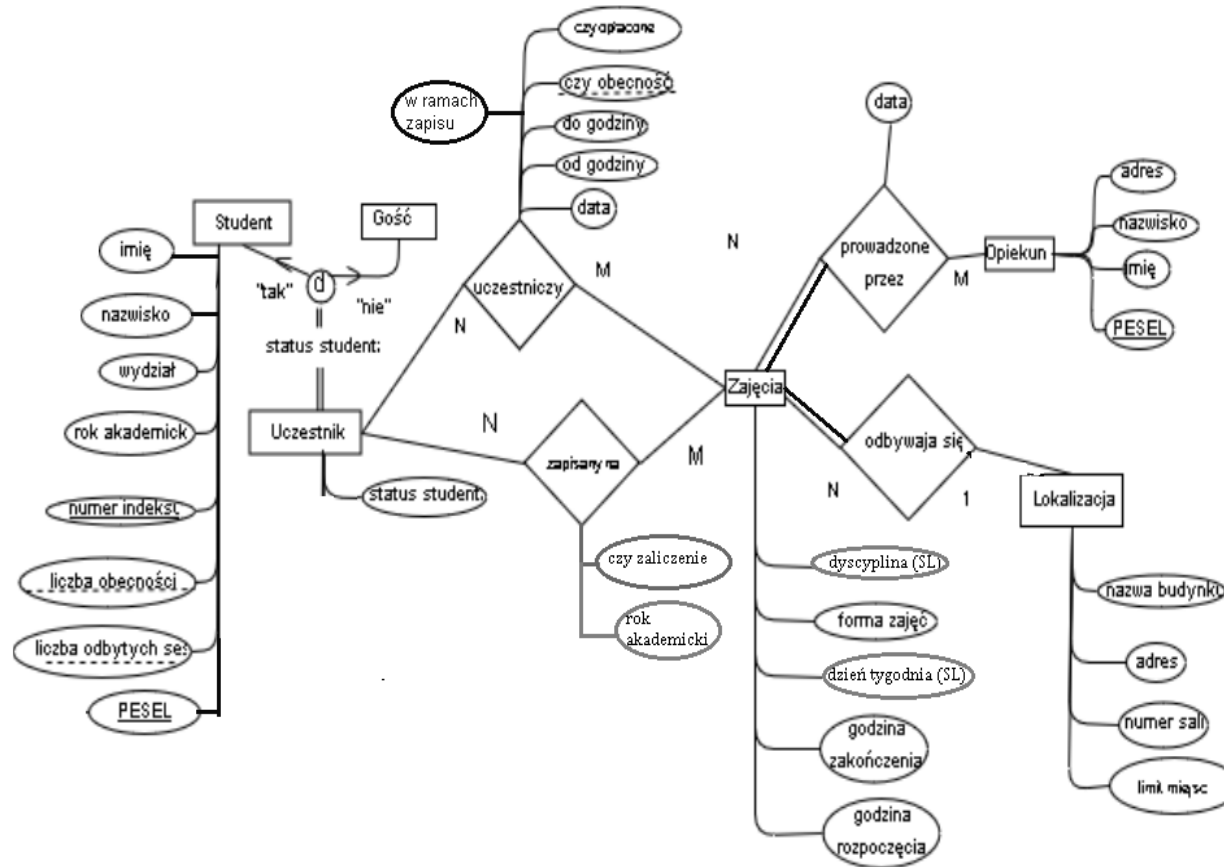
1 Założenia projektu - notatka

Wykonana aplikacja powinna realizować następujące założenia:

1. Każdy student powinien brać udział w zajęciach wychowania fizycznego, przy czym możliwy jest udział we więcej niż jednym rodzajów zajęć (przykładowo na piłkę nożną i judo). System przechowuje informację o roku akademickim studenta.
2. Każde zajęcia odbywają się w określonej lokalizacji (adres + numer sali) i liczba osób, które jednocześnie mogą brać udział w zajęciach jest z ograniczona.
3. Wyróżnia się dwa rodzaje zajęć sportowych:
 - (a) ci którzy wybrali daną dyscyplinę zapisują się do konkretnych grup i muszą odbywać zajęcia w swojej grupie zajęciowej (np dwie grupy judo 18:15 - 19:00 i 20:15 - 21:00),
 - (b) zajęcia otwarte dla wszystkich zapisanych na wybraną dyscyplinę (np basen od 8:00 do 20:00 we środę).
4. Każde zajęcia posiadają swojego opiekuna lub opiekunów. System powinien również przechowywać historię - kto w danym dniu i danej godzinie był opiekunem danej grupy.
5. Rejestrowany jest udział studenta w zajęciach: Aby uzyskać obecność na zajęciach typu a należy pojawić się na zajęciach ze swoją grupą. W przypadku zajęć typu rejestrowany jest czas wejścia i wyjścia studenta z zajęć - do obecności wymagane jest conajmniej 30 minut na zajęciach
6. System powinien obsługiwać także udział w zajęciach przez ludzi nie należących do uczelni lub studentów, którym nie przysługują już darmowe zajęcia sportowe. Biorą oni udział odpłatnie - należy skontrolować czy opłacili udział. Nie kontroluje się ich obecności, jednakże muszą posiadać konto w systemie i muszą się zalogować żeby np. przejrzeć listę zajęć.
7. Przewiduje się następujących użytkowników systemu:
 - (a) Student/klient, który może obejrzeć ofertę zajęć WF
 - (b) Portier, który rejestruje obecność studenta na zajęciach, dodając rekordy związane z czasem wejścia/wyjścia z zajęć

- (c) Administrator, który przypisuje studentów do zajęć, monitoruje liczbę obecności i wystawia zaliczenie zajęć WF na ich podstawie
- (d) Zwykły użytkownik bez specjalnych uprawnień, który po zalogowaniu się do systemu może przejrzeć listę zajęć.

2 Diagram związków encji



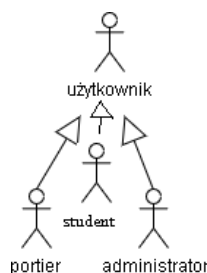
Rysunek 1: Model koncepcyjny bazy danych systemu eWF

3 Aktorzy i diagram przypadków użycia.

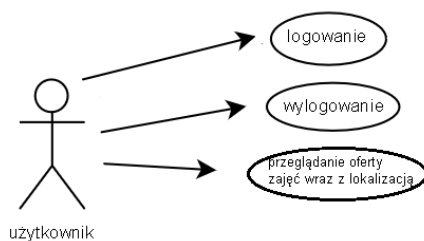
3.1 Opis aktorów i czynności, które mogą wykonywać.

Przewiduje się następujących aktorów systemu:

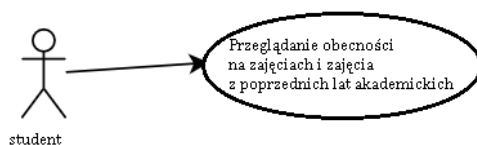
- Użytkownik, który może zalogować się do systemu i wylogować się z niego. Może także obejrzeć ofertę zajęć: dyscypliny, godziny zajęć, limity miejsc, lokalizację zajęć. Przewiduje się także trzy typy użytkownika, którzy posiadają szczególne uprawnienia:
 - Portier, który dodaje rekordy odnośnie przyścia studenta na zajęcia, opuszczenia zajęć przez studenta, a także usuwania/modyfikowania tych rekordów w przypadku popełnionego błędu.
 - Administrator systemu eWF, który dodaje/modyfikuje/usuwa zajęcia z oferty; dodaje nowe lokalizacje zajęć lub usuwa niedostępne już lokalizacje albo modyfikuje jeśli, lokalizacja którychś zajęć ulega zmianie. Administrator również przypisuje/zmienia opiekunów grup i odpowiada za zapisy studentów do grup (wraz z kontrolą opłacenia zajęć w przypadku zajęć płatnych). Co więcej, dodanie/usuwanie/edytowanie użytkowników należy do obowiązków administratora.
 - Student, który może zobaczyć swoje obecności w zajęciach i zajęcia na które uczęszczał w poprzednich latach akademickich.



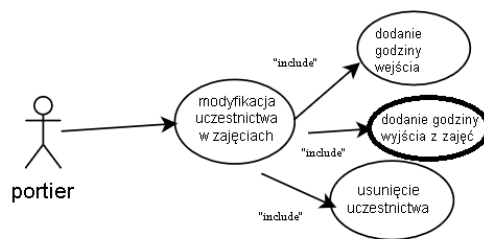
Rysunek 2: Relacje między aktorami.



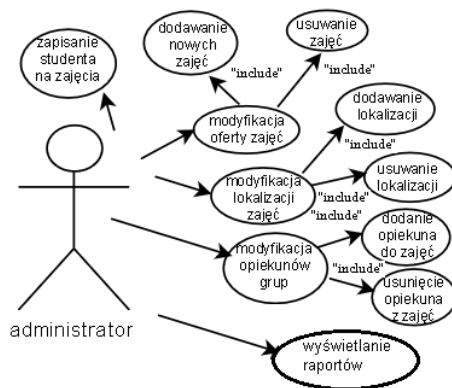
Rysunek 3: Możliwe operacje użytkownika



Rysunek 4: Możliwe operacje studenta

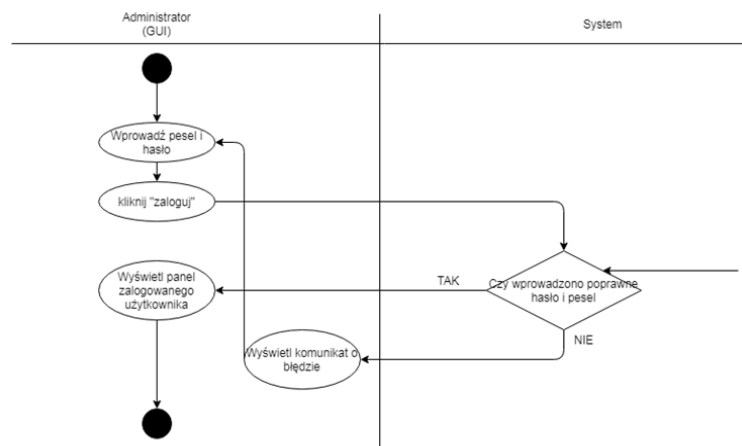


Rysunek 5: Możliwe operacja portiera

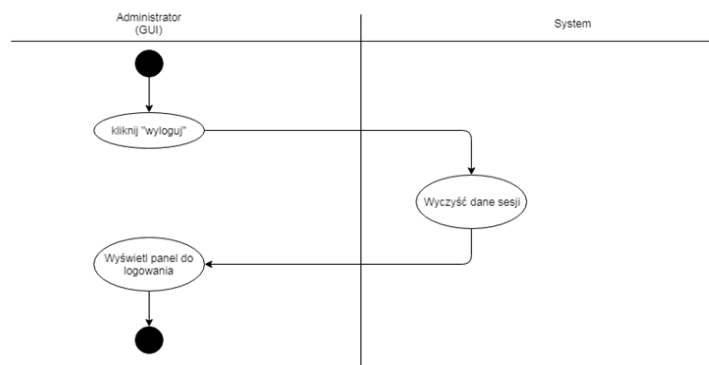


Rysunek 6: Możliwe operacje administratora

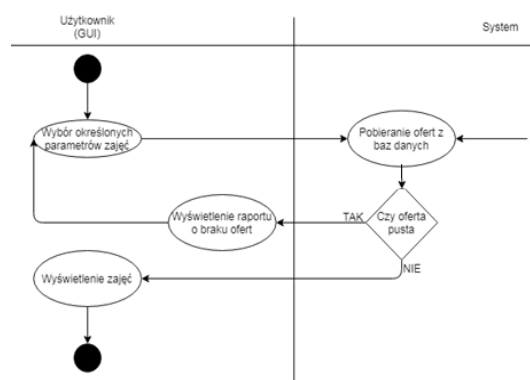
3.2 Przypadki użycia przez użytkownika.



Rysunek 7: Diagram czynności podczas logowania

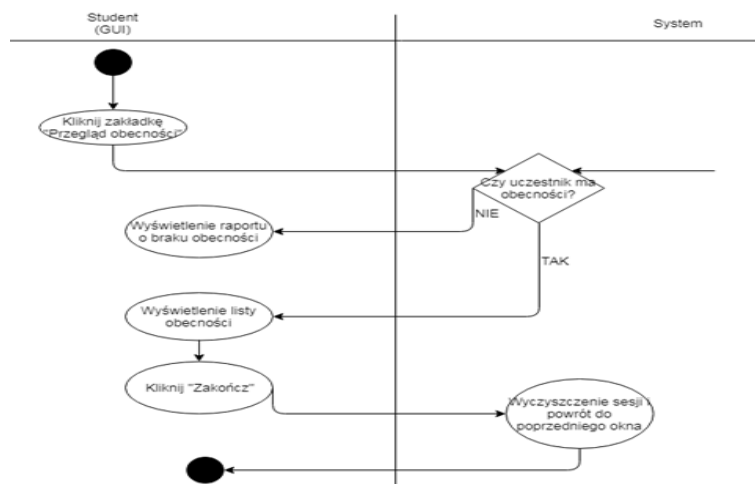


Rysunek 8: Diagram czynności podczas wylogowania



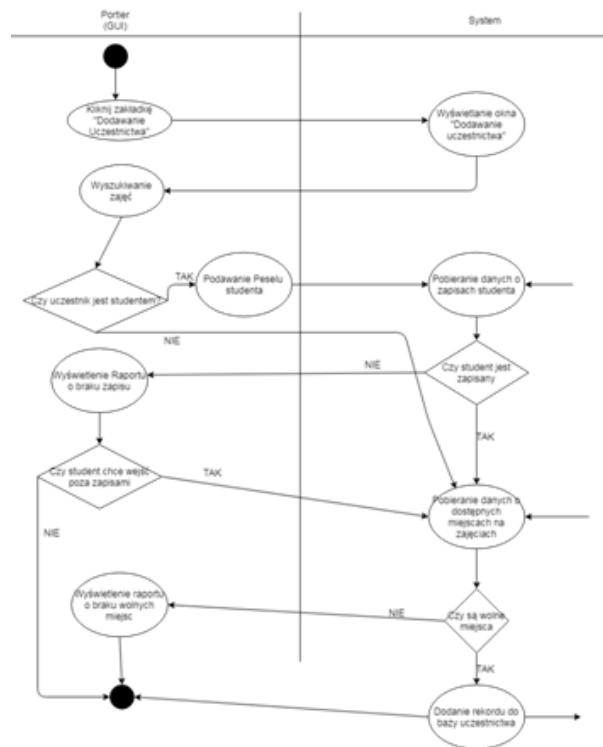
Rysunek 9: Diagram czynności podczas wyświetlania zajęć i ich lokalizacji

3.3 Przypadki użycia przez studenta.

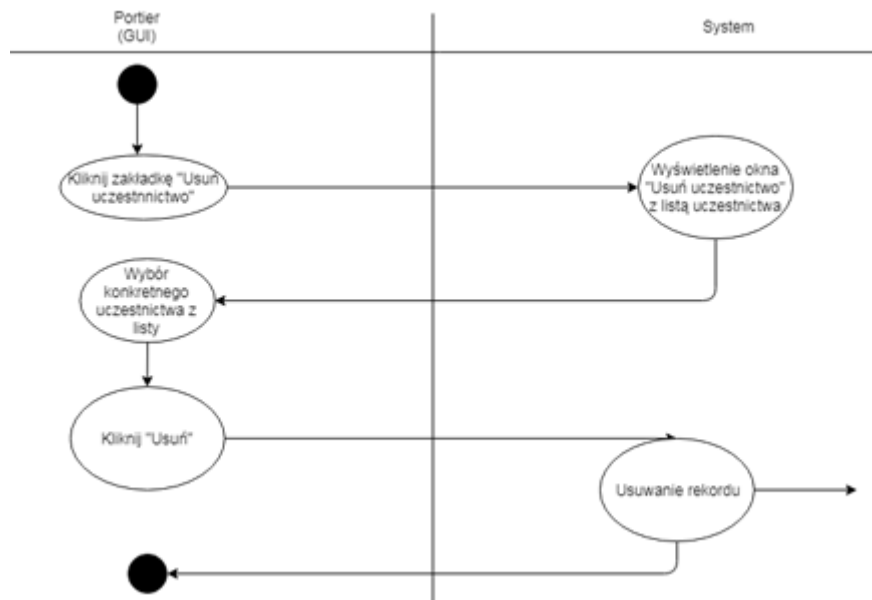


Rysunek 10: Diagram czynności podczas sprawdzania obecności przez studenta

3.4 Przypadki użycia przez portiera.

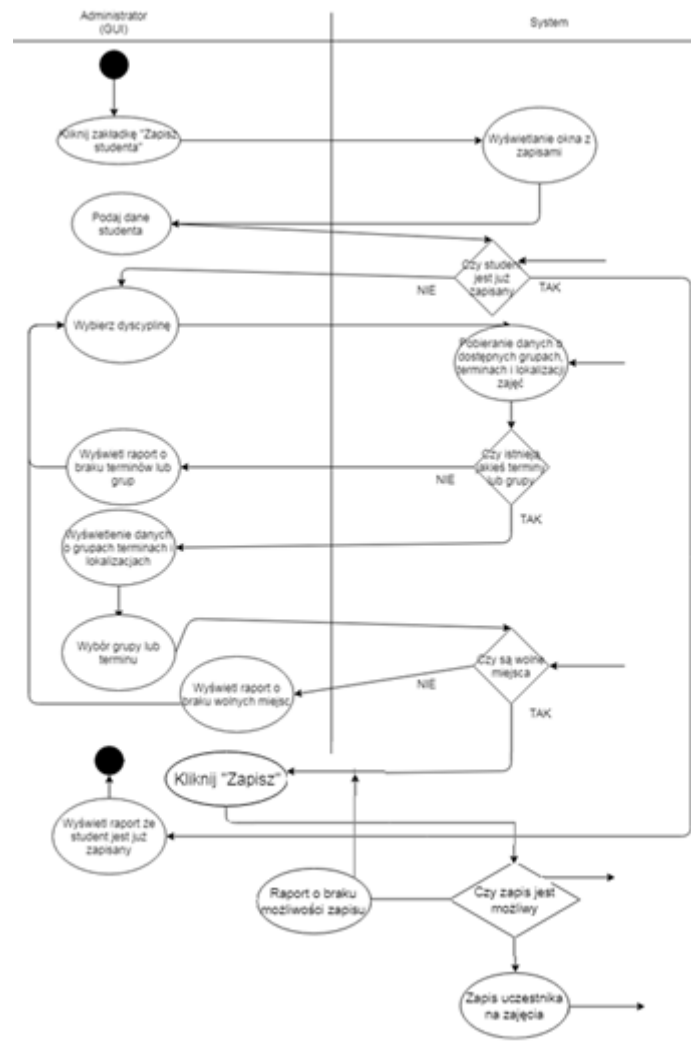


Rysunek 11: Diagram czynności podczas dodawania rekordu wejścia uczestnika na zajęcia

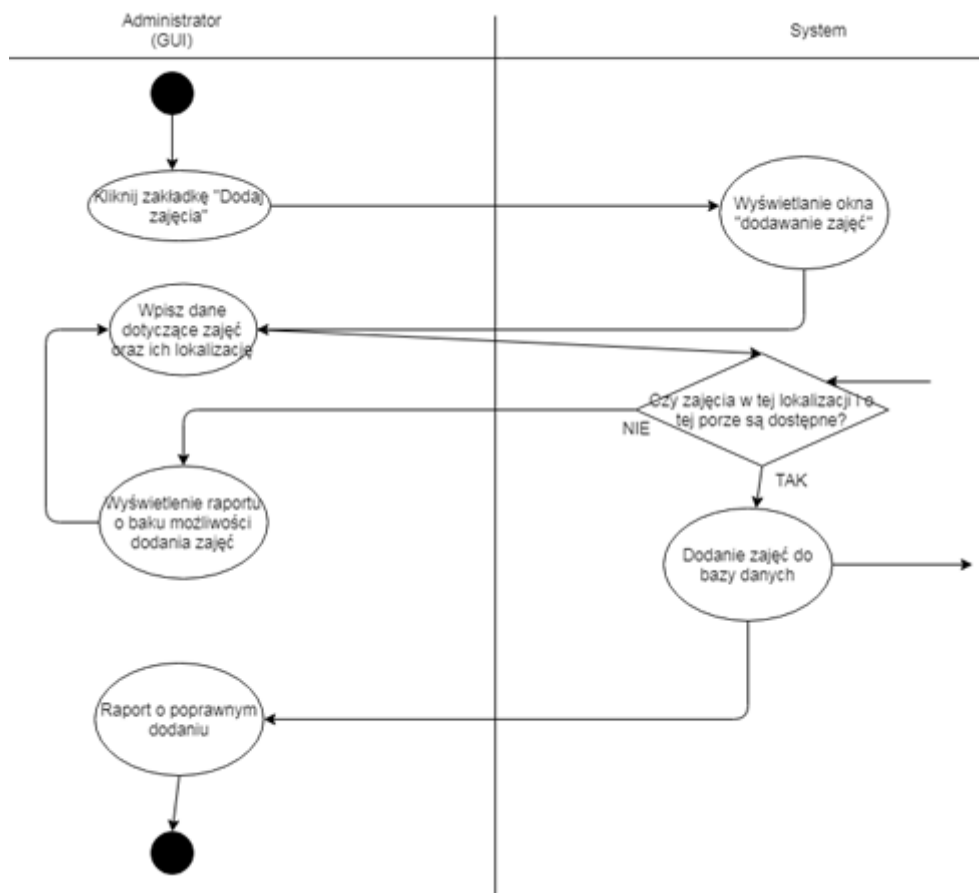


Rysunek 12: Diagram czynności podczas usuwania uczestnictwa w zajęciach

3.5 Przypadki użycia przez administratora.

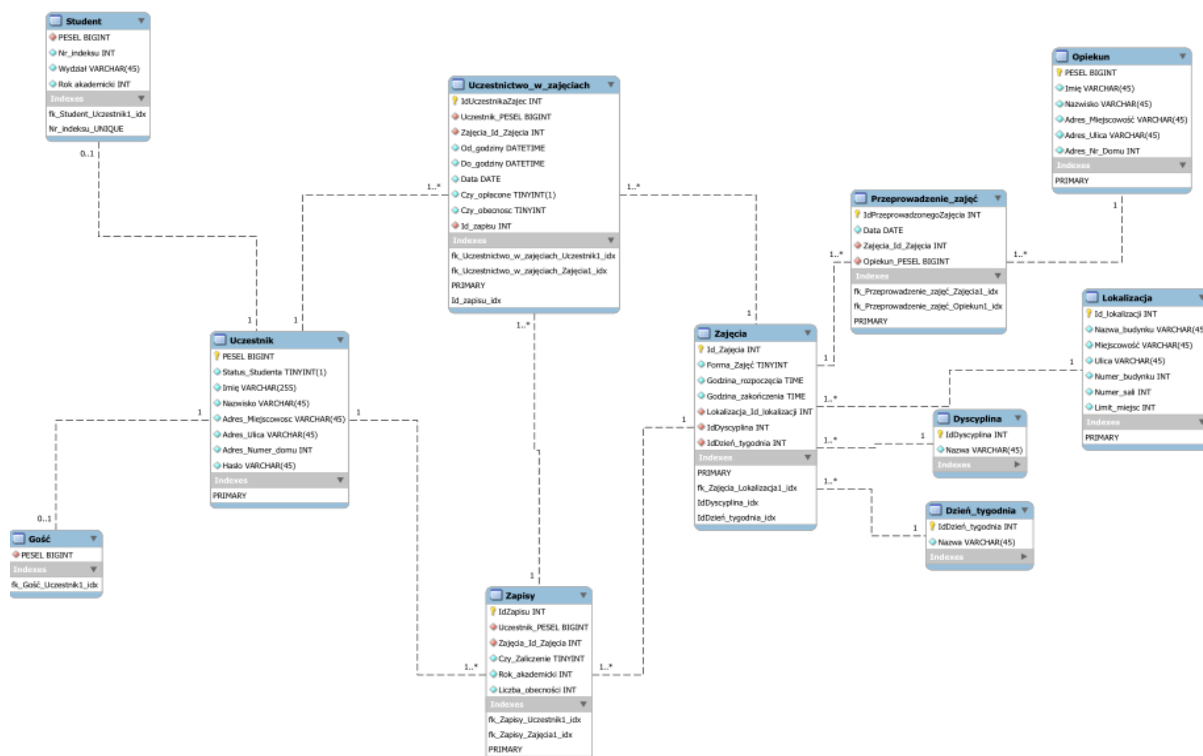


Rysunek 13: Diagram czynności podczas zapisywania studenta na zajęcia



Rysunek 14: Diagram czynności podczas dodawania nowych zajęć do oferty

4 Model relacyjny bazy danych.



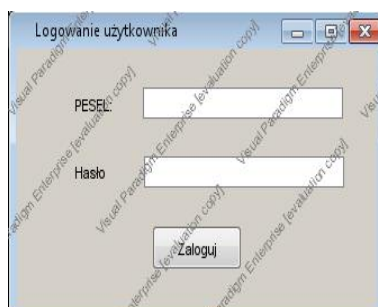
Rysunek 15: Model bazy danych w notacji UML

5 Projekt GUI i przejścia między ekranami.

Opisy przejść pomiędzy ekranami są zawarte w powyższych diagramach przepływu danych.

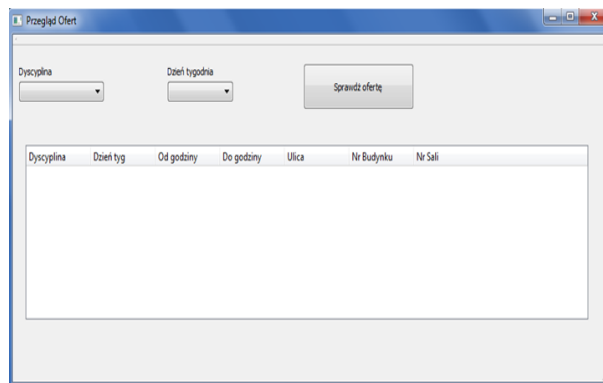
Projekt ekranów użytkowników:

5.1 Ekran użytkownika.



Rysunek 16: Ekran logowania użytkownika

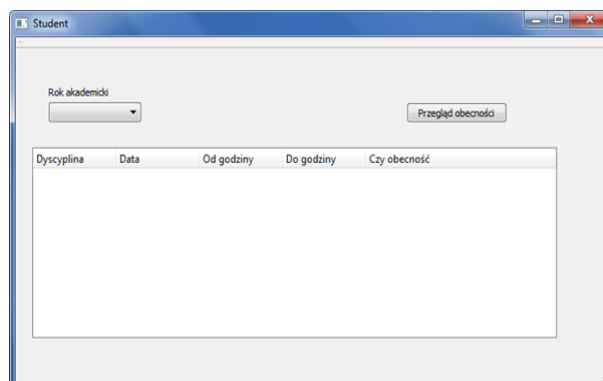
Podajemy Pesel i hasło, następnie aktywuje się "Zaloguj" i go naciskamy. Wyświetla się odpowiedni komunikat i zostajemy przekierowani do odpowiedniego okna.



Rysunek 17: Ekran przeglądania ofert zajęć i ich lokalizacji przez użytkownika

Wybieramy przynajmniej jeden parametr dyscyplina bądź dzień tygodnia, aktywuje się wtedy przycisk "Sprawdź ofertę". Po naciśnięciu wyświetla się nam lista ofert.

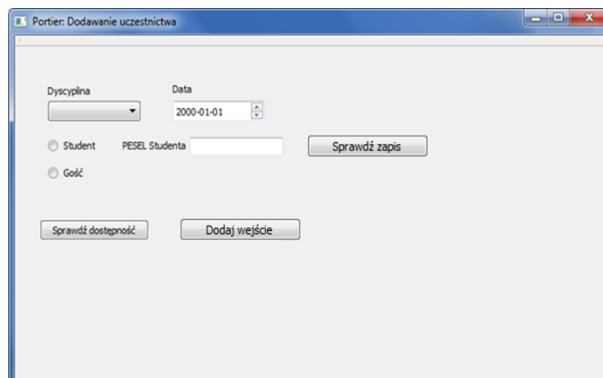
5.2 Ekran studenta.



Rysunek 18: Ekran przeglądania obecności na zajęciach

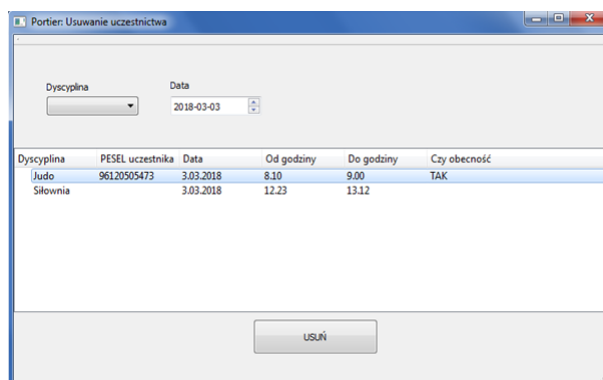
Wybieramy rok akademicki, aktywuje się wtedy przycisk przegląd obecności. Po naciśnięciu widzimy obecności studenta w danym roku akademickim.

5.3 Ekrany portiera.



Rysunek 19: Ekran dodawania wejście użytkownika na zajęcia

Na początku wybieramy dyscyplinę i datę. Potem zaznaczamy czy uczestnik jest to student czy gość. W przypadku wyboru studenta, wpisujemy jego Pesel, wtedy uaktywnia się nam przycisk "Sprawdź zapis". Naciskając go wyświetla się raport mówiący czy student jest zapisany czy nie i jeśli nie jest to czy chce wejść poza zapisami. Jeśli tak to sprawdzamy dostępność, dla gościa tak samo. Jeśli są miejsca to naciskamy dodaj wejście.



Dyscyplina	PESEL uczestnika	Data	Od godziny	Do godziny	Czy obecność
Judo	96120505473	3.03.2018	8.10	9.00	TAK
Silownia		3.03.2018	12.23	13.12	

Rysunek 20: Ekran usuwania uczestnictwa w zajęciach

Po wyborze dyscypliny i daty wyświetla nam się lista uczestnictw wtedy wybieramy konkretne, aktywuje się przycisk usuń i naciskamy go w celu usunięcia uczestnictwa

5.4 Ekran administratora.

Data	Nowa kolumna	Do godziny	Nowa kolumna	Nr budynku
------	--------------	------------	--------------	------------

Rysunek 21: Ekran zapisywania studenta na zajęcia

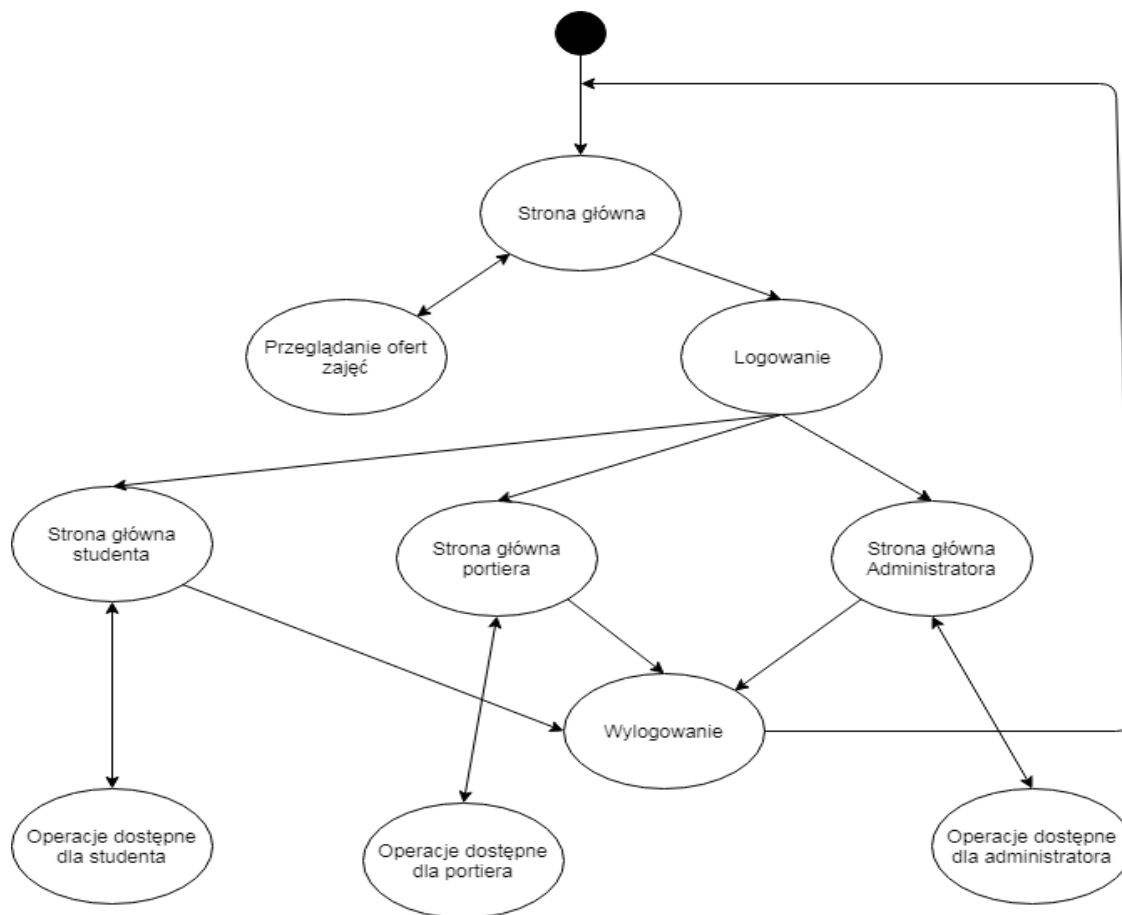
Na początku wprowadzamy dane studenta. Następnie aktywuje się przycisk "Sprawdź czy student zapisany". Jeśli student nie jest jeszcze zapisany to wybieramy dyscyplinę na jaką się chce zapisać student. Wyświetla nam się lista ofert. Wtedy wybieramy konkretną z listy i naciskamy "Sprawdź dostępność zajęć". Jeśli wszystko jest ok, aktywuje się przycisk "Zapisz studenta" Po naciśnięciu następuje zapisanie lub odpowiedni komunikat.

Zajęcia	Lokalizacja
---------	-------------

Rysunek 22: Ekran dodawania nowych zajęć do oferty

Na początku uzupełniamy dane dotyczące zajęć i wybieramy ich lokalizację. Po wyborze i uzupełnieniu wszystkich danych następuje aktywacja "Sprawdź dostępność zajęć". Po naciśnięciu wyskakuje komunikat o braku możliwości lub w przeciwnym razie aktywuje się przycisk "Dodaj zajęcia". Po naciśnięciu zajęcia zostają dodane do bazy danych.

5.5 Diagram przejść pomiędzy ekranami.



Rysunek 23: Diagram przejść pomiędzy ekranami.

6 Raporty.

6.1 Najpopularniejsze zajęcia.

Raport podaje liczbę zapisów na dane zajęcia oraz procentowy udział tych zapisów we wszystkich zapisach, wyniki mogą być przedstawione również w formie wykresu, a raport dotyczy jednego roku akademickiego.

Dane wyświetlane:

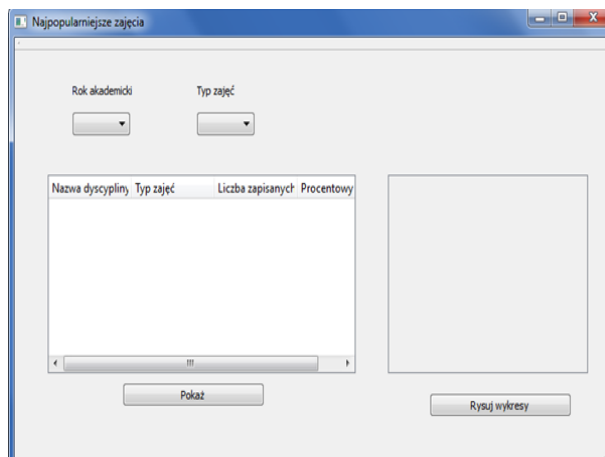
- nazwa dyscypliny
- typ zajęć
- liczbę zapisanych studentów
- procentowy udział zapisanych studentów

Parametry filtracji:

- rok akademicki
- typ zajęć (otwarte lub w grupowe)

Możliwości sortowania:

- po liczbie zapisów
- alfabetycznie po nazwach dyscyplin



Rysunek 24: Projekt GUI raportu "najpopularniejsze zajęcia"

6.2 Godziny szczytu.

Raport podaje w jakich godzinach jest ilu uczestników na zajęciach typu "otwarte". Zakładamy, że raport dotyczy jednego miesiąca.

Dane do wyświetlenia:

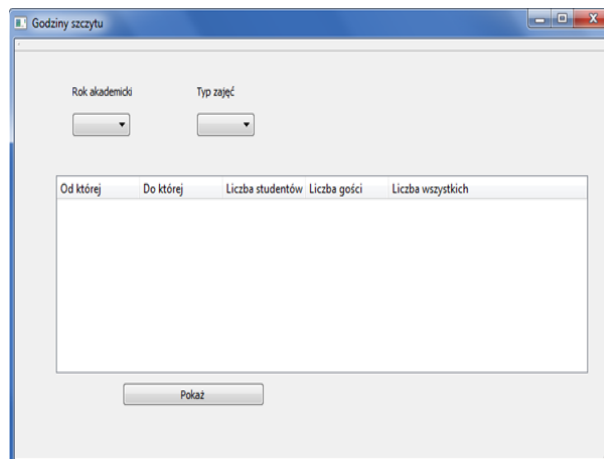
- przedział czasowy (np. 13-14) co godzinę
- liczba studentów
- liczba gości
- suma wszystkich uczestników

Parametry filtracji:

- wybór dyscypliny
- wybór miesiąca

Możliwości sortowania:

- po liczbie wszystkich uczestników
- po godzinach



Rysunek 25: Projekt GUI raportu "godziny szczytu"

6.3 Zaliczenia.

Raport przedstawia liczbę zaliczeń z danej dyscypliny, biorąc pod uwagę jeden rok akademicki. Dane do wyświetlenia:

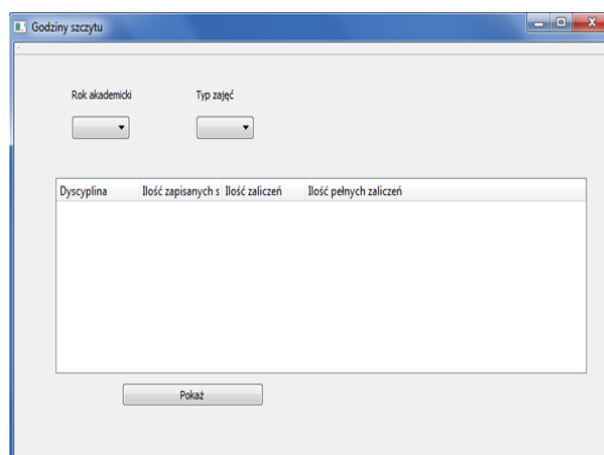
- nazwa dyscypliny
- ilość zapisanych uczestników
- ilość zaliczeń
- ilość zaliczeń z pełną liczbą obecności

Parametry filtracji:

- typ zajęć (otwarte lub grupowe)
- rok akademicki

Możliwości sortowania:

- po liczbie zaliczeń
- alfabetycznie po dyscyplinie



Rysunek 26: Projekt GUI raportu "zaliczenia"

7 Implementacja systemu bazodanowego.

Do stworzenia interfejsu graficznego wykorzystano język C++ oraz wbudowane narzędzia do projektowania dostępne w środowisku QtCreator. GUI połączono z lokalną bazą danych uzupełnioną przykładowymi danymi. Zaimplementowano wszystkie raporty oraz część ekranów.

7.1 Zaimplementowane ekrany

Do zaimplementowania wybrano następujące ekrany:

- Ekran logowania użytkownika,
- Ekran przeglądania ofert zajęć przez użytkownika,
- Ekran dodawania uczestnictwa w zajęciach dla portiera,
- Ekran zapisywania studenta na zajęcia dla administratora.

Poza tym zaimplementowano strony służące do nawigowania po programie:

- Stronę główną użytkownika,
- Stronę główną studenta,
- Stronę główną portiera,
- Stronę główną administratora.

7.2 Zaimplementowane raporty

Zaimplementowano wszystkie raporty przewidziane w projekcie. Poniżej przedstawiono zapytania w języku SQL wykorzystane w tych raportach.

- Najpopularniejsze zajęcia:

```
"SELECT d.Nazwa, zaj.Forma_zajęć, COUNT(d.nazwa) AS 'Liczba zapisanych',
count(d.Nazwa)/(select COUNT(*) FROM zapisy where zapisy.
Rok_akademicki = " + ui->year_box->currentText() + ")*100 AS '
Procent wszystkich zapisów' FROM zajęcia as zaj join dyscyplina as d
on zaj.idDyscypliny = d.idDyscypliny join zapisy as zap on zaj.
Id_Zajęcie = zap.Id_Zajęcie where zap.Rok_akademicki = " + ui->year_box
->currentText() + qry_end + " group by d.Nazwa"}
```

ui->year_box->currentText() zawiera rok wybrany przez użytkownika
qry_end jest to opcjonalny warunek formy zajęć

- Godziny szczytu:

```
"SELECT godziny.Od, godziny.Do, COUNT(godziny.Od) AS Wszyscy, sum(case
when u.Czy_opłacone = 1 then 1 else 0 end) Goście, sum(case when u.
Czy_opłacone = 0 then 1 else 0 end) Studenci FROM
uczestnictwo_w_zajęciach as u JOIN zajęcia as zaj on zaj.Id_Zajęcie =
u.Id_Zajęcie join dyscyplina as d on d.idDyscypliny = zaj.idDyscypliny
join godziny where u.Od_godziny <= godziny.Do AND u.Do_godziny >=
godziny.Od and year(u.Data) = " + ui->year_box->currentText() + "
and month(u.Data) = " + ui->month_box->currentText() + " and d.Nazwa
= '" + ui->sport_box->currentText() + "' GROUP BY godziny.Od"
```

ui->year_box->currentText() zawiera rok wybrany przez użytkownika

ui->month_box->currentText() zawiera miesiąc wybrany przez użytkownika
 ui->sport_box->currentText() zawiera wybraną przez użytkownika
 dyscyplinę

- Zaliczenia:

```
"select q1.dyscyplina AS 'Dyscyplina', sum(q2.lz) AS 'Liczba zapisanych',
sum(q1.lzal) AS 'Liczba zaliczeń', sum(q1.lzpo) AS 'Liczba
zaliczeń z pełną obecnością' FROM (select zaj.id_zajęc as i, d.Nazwa
as 'Dyscyplina', sum(case when zap.Czy_zaliczenie = 1 then 1 else 0
end) 'Lzal', sum(case when zap.Liczba_obecności = q.lpo then 1 else
0 end) 'Lzpo' from dyscyplina AS D JOIN zajęcia AS ZAJ ON D.
idDyscypliny = ZAJ.idDyscypliny join zapisy as zap on zap.id_zajęc =
zaj.id_zajęc join (SELECT zaj.Id_Zajęc as idz, dyscyplina.Nazwa,
COUNT(p.date) as lpo FROM przeprowadzenie_zajęc as p join zajęcia as
zaj on zaj.Id_Zajęc = p.Id_Zajęc join dyscyplina on zaj.
idDyscypliny = dyscyplina.idDyscypliny group by zaj.Id_Zajęc) as q
on q.idz = zaj.Id_Zajęc where zap.rok_akademicki = " + ui->year_box
->currentText() + qry_end + " and zap.czy_zaliczenie = 1 group by
zaj.Id_zajęc) AS q1 JOIN (SELECT zaj.id_zajęc AS i, d.Nazwa AS '
Dyscyplina', COUNT(d.nazwa) AS 'Lz' FROM zajęcia as zaj join
dyscyplina as d on zaj.idDyscypliny = d.idDyscypliny join zapisy as
zap on zaj.Id_Zajęc = zap.Id_Zajęc where zap.Rok_akademicki = " + ui
->year_box->currentText() + qry_end + " group by zaj.id_zajęc\) as
q2 on q1.i = q2.i group by Dyscyplina"
```

ui->year_box->currentText() zawiera rok wybrany przez użytkownika
 qry_end jest to opcjonalny warunek formy zajęć