

Sprawozdanie z przedmiotu Podstawy Teleinformatyki

Temat: Sprawdzanie obecności za pomocą legitymacji studenckiej.

Wykonane przez:

Parus Michał
Kozioł Aleksander

Uzasadnienie wyboru tematu projektu.

Temat pracy zaliczeniowej z przedmiotu "Podstawy Teleinformatyki", został przez nas wybrany z powodu kilku czynników:

- a. Użyteczność - sprawdzanie obecności na wykładach lub zajęciach laboratoryjnych jest wielkim krokiem w stronę komputeryzacji całego systemu edukacji. Już teraz legitymacja studencka posiada wiele różnych funkcji i używanie jej do sprawdzania obecności jest kolejnym nasuwającym się krokiem. Wielu wykładowców wciąż korzysta z papierowych listy obecności (czy to w kalendarzach czy na luźnych kartkach), co nie jest pewnym źródłem informacji i bardzo podatnym na oszustwa.
- b. Oszczędność czasu - rozwiązanie przez nas zaproponowane może zaoszczędzić wiele czasu, który zamiast poświęcać na wpisywanie się na wszelkie listy obecności można poświęcić na dokładniejsze prowadzenie zajęć i wykładów.
- c. Jednolity system zarządzania - sprawdzanie obecności za pomocą legitymacji studenckiej, może być sprawdzane i kontrolowane przez jeden system. Jest to rozwiązanie, które zastąpi i uporządkuje wszelkie papierowe listy obecności, zajmujące kolejne kartki w teczkach wykładowców. Ulepszy to również przepływ informacji pomiędzy wykładowcami a studentami - student może bardzo łatwo sprawdzić na ilu wykładach już był, lub wykładowca może bezproblemowo sprawdzać listy studentów z poprzednich zajęć
- d. Duża baza danych - dzięki aplikacji systemowej do przedstawionego systemu można w bardzo prosty sposób sprawdzać listy obecności z poprzednich miesięcy, lub nawet lat - bez konieczności szukania potrzebnej ku temu kartki z listą.
- e. Doświadczenie projektowe - temat prezentacji był już przez nas podejmowany w przeszłości lecz nie został w 100% ukończony. Dlatego też obecnie tego projektu nie musieliśmy wykonywać o zera, a mieliśmy już podstawy całego oprogramowania

Podział obowiązków:

Michał Parus:

- A. Stworzenie biblioteki służącej do komunikacji czytnika legitymacji studenckich na płycie Raspberry Pi, a serwerem gdzie był trzymany cały program
- B. Stworzenie biblioteki służącej do komunikacji czytnika legitymacji studenckich na płycie Raspberry Pi, a serwerem gdzie był trzymany cały program
- C. Wykonanie aplikacji klienckiej z widoku studenta, wykładowcy oraz administratora systemu.
 - a. Widok studenta, pozwala na sprawdzenie dawnych wykładach i obecności oraz przeglądanie całego planu semestru w widoku kalendarza.
 - b. Widok wykładowcy pozwala na kontrolę obecności na zajęciach. Aplikacja pozwala również na zdalne wprowadzenie obecności danego studenta (np w wypadku braku legitymacji studenckiej)
 - c. Widok administratora pozwala na wiele rzeczy, m.in.:
 - i. Dodawanie nowych studentów do bazy danych wraz z przypisanymi do nich legitymacjami studenckimi
 - ii. Edytowanie danych studenta
 - iii. Przeglądanie i kontrolowanie odczytów z legitymacji
 - iv. Dodawanie wykładowców do bazy danych i edytowanie (np dodawanie nowych przedmiotów dla wykładowcy)
 - v. Przypisywanie zajęć z danego przedmioty do prowadzących zajęcia
 - vi. Tworzenie nowych roczników oraz kierunków
- D. Konfiguracja bazy danych Mysql oraz utworzenie odpowiednich tabel dla poprawnego działania systemu.

Aleksander Kozioł:

- Testowanie działania aplikacji klienckiej
 - Wprowadzania usprawnień do działania aplikacji oraz pomoc w projektowaniu systemu
 - Wspólnie z Michałem Parusem stworzenie całej dokumentacji projektowej
- I. Założenia projektu:
- każdy z wykładowców ma dostęp, poprzez aplikację, do listy zajęć, które prowadzi oraz przypisanych do nich studentów, gdzie może sprawdzić ilu studentów jest obecnych na zajęciach,
 - wykładowca ma możliwość ręcznego dodania obecności danemu studentowi,

- studenci po wejściu do sali mogą uzyskać obecność po przyłożeniu legitymacji do czytnika,
- administrator ma możliwość dodawania nowych wykładowców jak i również nowych studentów,

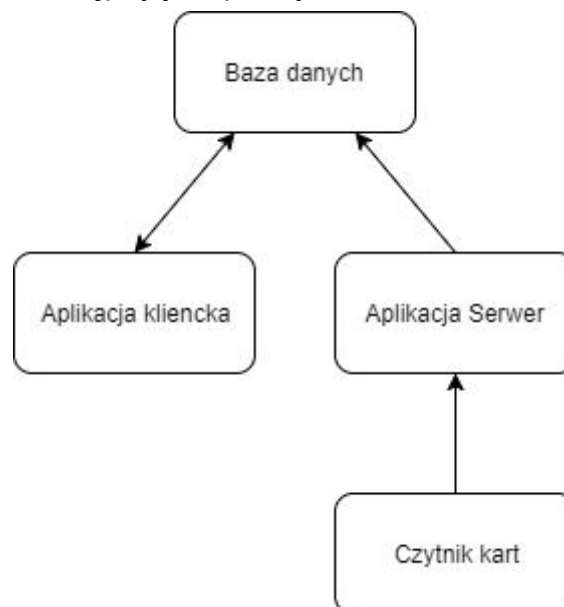
II. Ogólny zarys systemu:

W projekcie mamy trzy główne elementy, które razem tworzą cały system:

- czytnik legitymacji studenckich,
- aplikacja serwerowa łącząca czytnik z bazą danych,
- panel sterowania dla administratora oraz wykładowców.

Czytnik legitymacji studenckich działa na platformie Raspberry Pi z uruchomionym systemem Linux Raspbian Lite. Skrypt obsługujący odczytywanie informacji o przyłożonej legitymacji napisany jest w języku Python. Czytnik łączy się z serwerem napisanym w języku C# poprzez protokół TCP, wysyłając informacje o numerze UID zeskanowanej karty. Serwer przyjmujący informacje o numerze karty sprawdza czy w bazie danych jest dany użytkownik, jeżeli nie to dodajemy go jako nieznanego oraz sprawdzamy czy zapisany użytkownik ma w tym momencie jakieś zajęcia, na których może uzyskać obecność. Aplikacja kliencka podzielona jest na dwa widoki: dla wykładowców oraz dla administratora. Całość napisana jest w języku C# w oparciu o Windows Forms oraz zapytania SQL.

Cały projekt do wglądu dostępny jest pod tym [linkiem](#).



Schemat całego systemu.

III. Działanie systemu:

1. Czytnik kart

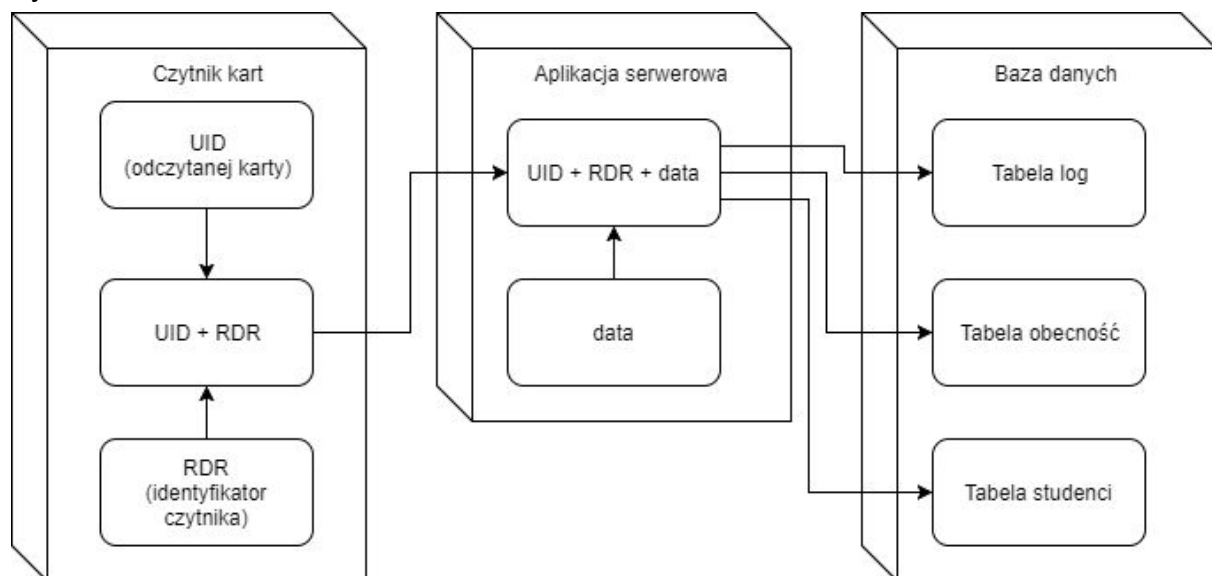
Do platformy Raspberry Pi przyłączona jest moduł RFID skanujący legitymację studencką i pobierający z niej, unikatowy dla każdej karty, numer identyfikacyjny

(UID) i wysyła go do serwera razem z identyfikatorem czytnika (RDR), który przetwarza dane dalej.



2. Aplikacja serwerowa:

Aplikacja odbiera dane wysłane przez czytnik oraz dodaje do nich datę. Dodatkowo sprawdzane jest czy zeskanowana karta znajduje się na liście w bazie danych, jeżeli nie rozpoznano jej to jest ona dodawana do listy, aby potem można było przypisać ją do danego studenta. Każdy pakiet przychodzący zapisywany jest w tabeli log dla archiwizacji. Sprawdzamy również czy możemy dodać studentowi obecność na wykładzie.



3. Aplikacja kliencka:

Najbardziej rozbudowany element całego systemu. Aplikacja podzielona jest na dwa oddzielne widoki: dla administratora oraz dla wykładowców. Możliwości jakie zapewnia nam widok administratora:

- dodawanie, usuwanie i edycja studentów,
- dodawanie, usuwanie i edycja roczników,

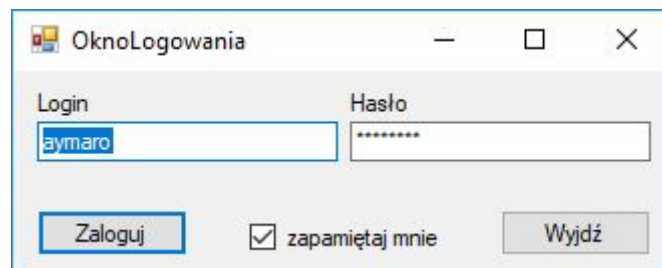
- przypisywanie studentów do roczników,
- podgląd log'u,
- sprawdzenie listy przedmiotów,
- zarządzanie wykładowcami oraz przedmiotami, które wykładają.

Z kolei widok dla wykładowców obejmuje:

- wybór daty i pokazanie wykładów tego dnia,
- dodanie nowego wykładu z podaniem daty, przedziału godzin oraz rocznika,
- przegląd studentów, którzy powinni być na danym wykładzie oraz sprawdzenie ich obecności,
- zmiana obecności poszczególnych studentów.

IV. Prezentacja systemu:

Po uruchomieniu aplikacji klienta ukazuje nam się okno logowania



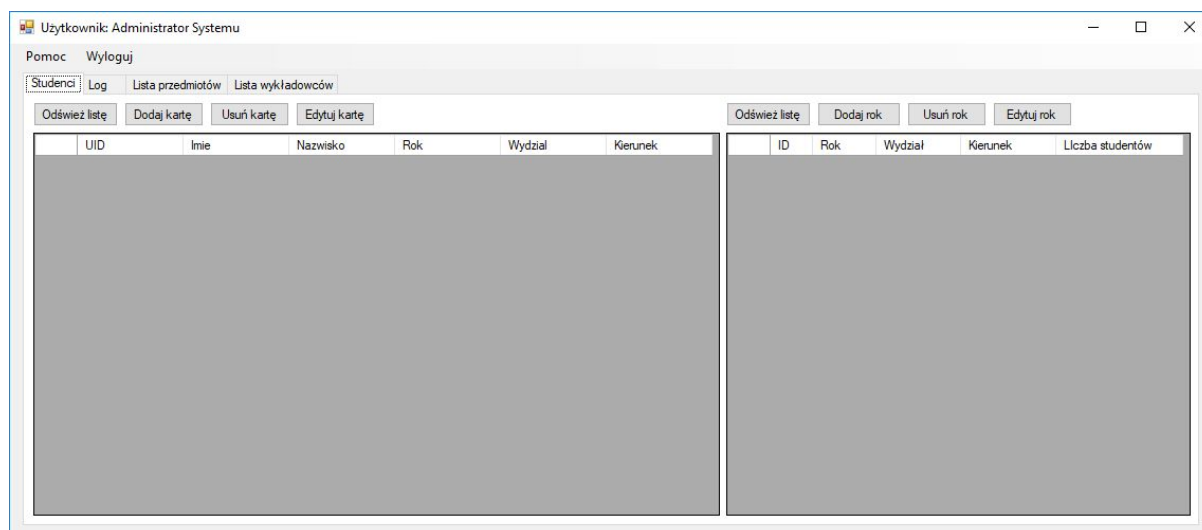
Po wpisaniu danych system sprawdzi czy dany użytkownik istnieje. W zależności czy konto jest wykładowcy czy też administratora system otworzy różne widoki.

Kod sprawdzający użytkowników:

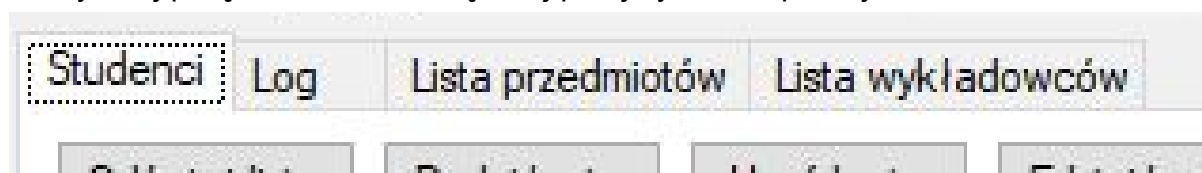
```
string query = String.Format("SELECT ID, Imie, Nazwisko FROM wykładowcy  
WHERE login = '{0}' and haslo = '{1}'", login, password);
```

Jeżeli użytkownik taki istnieje i hasło się zgadza to polecenie zwróci on nam go z listy. Dodatkowo hasło jest przechowywane w formie skrótu MD5 także przed wysłaniem zapytania do bazy danych jest ono poddawane obróbce w funkcji skrótu.

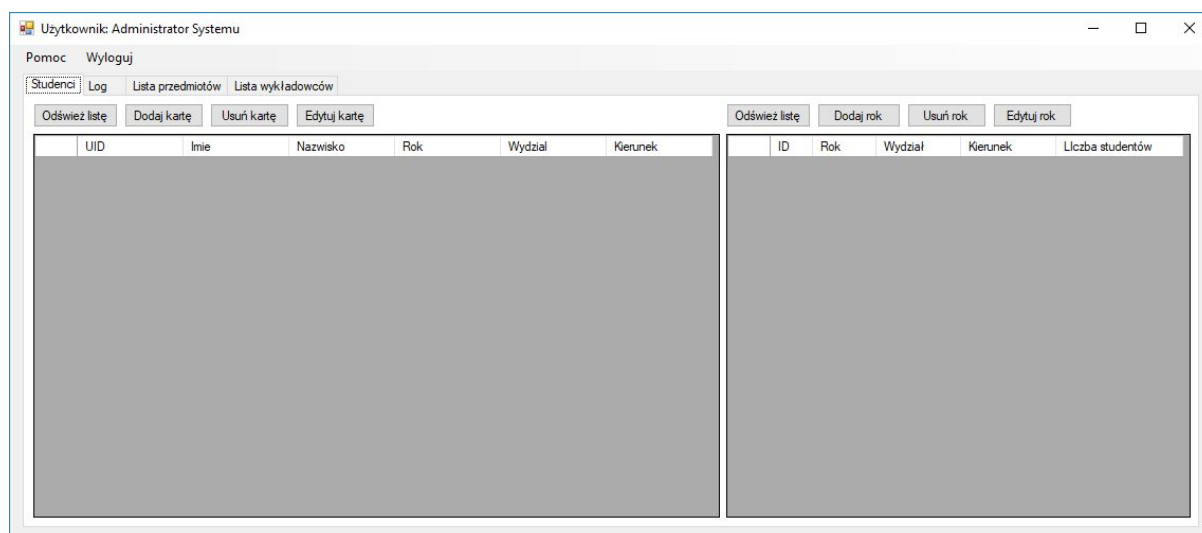
Podany użytkownik jest administratorem, także pokażę nam się takie okno:



Mamy tutaj parę zakładek dla większej przejrzystości aplikacji:



Pierwsza zakładka to widok studentów oraz roczników:



Po odświeżeniu obu list ukażą się nam wszyscy studenci oraz roczniki:

Uzytkownik: Administrator Systemu

Pomoc Wyloguj

Studenci Log Lista przedmiotów Lista wykładów

Odśwież listę Dodaj kartę Usuń kartę Edytuj kartę

	UID	Imię	Nazwisko	Rok	Wydział	Kierunek
▶	60DE5135	Michał	Parus	0	brak	brak
	FE8EB2C5	Aleksander	Kozioł	0	brak	brak
	D0C95335	Marta	Budna	0	brak	brak
	9BC355E3	Aleksander	Kozioł	0	brak	brak
	DBE254E3	Oskar	Masłek	2014	Elektryczny	Infomatyka
	FBEA54E3	Michał	Parus	2014	Elektryczny	Infomatyka

Odśwież listę Dodaj rok Usuń rok Edytuj rok

	ID	Rok	Wydział	Kierunek	Liczba studentów
▶	1	0	brak	brak	4
	10	2014	Elektryczny	Infomatyka	2

W tym momencie podwójnym kliknięciem możemy dodawać studentów do wybranych roczników. Dodatkowo wybierając rocznik widzimy studentów należących do danego rocznika:

Uzytkownik: Administrator Systemu

Pomoc Wyloguj

Studenci Log Lista przedmiotów Lista wykładów

Odśwież listę Dodaj kartę Usuń kartę Edytuj kartę

	UID	Imię	Nazwisko	Rok	Wydział	Kierunek
▶	60DE5135	Michał	Parus	0	brak	brak
	FE8EB2C5	Aleksander	Kozioł	0	brak	brak
	D0C95335	Marta	Budna	0	brak	brak
	9BC355E3	Aleksander	Kozioł	0	brak	brak
	DBE254E3	Oskar	Masłek	2014	Elektryczny	Infomatyka
	FBEA54E3	Michał	Parus	2014	Elektryczny	Infomatyka

Odśwież listę Dodaj rok Usuń rok Edytuj rok

	ID	Rok	Wydział	Kierunek	Liczba studentów
	1	0	brak	brak	4
▶	10	2014	Elektryczny	Infomatyka	2

Kod odpowiadający za pobieranie listy studentów oraz roczników:

Studenci:

```
string query = "SELECT a.UID, a.Imię, a.Nazwisko, b.rok, b.wydział, b.kierunek
FROM studenci a JOIN roczniki b ON a.rocznik = b.id";
```

Roczniki:

```
string query = "SELECT id, rok, wydział, kierunek, liczba FROM roczniki";
```

Oraz kod odpowiadający za zmianę rocznika u studenta:

```
string query = String.Format("UPDATE studenci SET rocznik='{0}' WHERE
UID='{1}'", rocznik, UID);
```


W każdej chwili możemy dodać studenta lub rocznik poprzez specjalne okna które wyglądają następująco:

The image shows two windows side-by-side. The left window, titled 'Dodaj kartę', has three input fields for 'UID', 'Imię', and 'Nazwisko', and a 'Powtórz hasło' label. It includes a checkbox for 'edytuj wpis' and buttons for 'Anuluj' and 'Dodaj'. The right window, titled 'Dodaj rocznik', has input fields for 'Id', 'Rok', a dropdown for 'Wydział', and a dropdown for 'Kierunek'. It also includes a checkbox for 'edycja wpisu' and buttons for 'Dodaj' and 'Anuluj'.

Po wypełnieniu wszystkich pól i zatwierdzeniu wykonuje się następujący kod:

Studenci:

```
string query = String.Format("INSERT INTO studenci (UID, Imie, Nazwisko, rocznik)
SELECT '{0}', '{1}', '{2}', '{3}' FROM studenci WHERE NOT EXISTS (SELECT *
FROM studenci WHERE UID='{0}') LIMIT 1 ;", UID, Imie, Nazwisko, "1");
```

Jest tutaj zabezpieczenie dzięki któremu nie możemy dodać dwóch studentów z tą samą kartą.

Roczniki:

```
string query = String.Format("INSERT INTO roczniki (rok, wydzial, kierunek)
VALUES ('{0}', '{1}', '{2}')", Rok, Wydzial, Kierunek);
```

Następną zakładką jest log, gdzie mamy zapisane wszystkie odczyty karty:

The screenshot shows the 'Log' tab selected in the application. It displays a table with the following columns: ID, Data, UID, RDR, Imie, and Nazwisko. The table contains 13 rows of data. Below the table, there is a checkbox for 'Odśwież listę' and a dropdown for 'Liczba wierszy' set to 100.

ID	Data	UID	RDR	Imie	Nazwisko
67	04.11.2017 00:2...	fe9eb2c5	1	Aleksander	Kozioł
68	04.11.2017 00:2...	60de5135	1	Michał	Parus
69	04.11.2017 00:2...	d0c95335	1	Marta	Budna
70	04.11.2017 00:2...	d0c95335	1	Marta	Budna
71	04.11.2017 00:2...	fe9eb2c5	1	Aleksander	Kozioł
72	04.11.2017 00:2...	fe9eb2c5	1	Aleksander	Kozioł
73	04.11.2017 00:2...	60de5135	1	Michał	Parus
74	04.11.2017 00:3...	d0c95335	1	Marta	Budna
75	04.11.2017 00:3...	fe9eb2c5	1	Aleksander	Kozioł
76	04.11.2017 00:3...	fe9eb2c5	1	Aleksander	Kozioł
77	04.11.2017 00:3...	fe9eb2c5	1	Aleksander	Kozioł
78	04.11.2017 00:4...	60de5135	1	Michał	Parus
79	04.11.2017 00:4...	60DE5135	1	Michał	Parus
80	04.11.2017 00:4...	60DE5135	1	Michał	Parus
81	04.11.2017 00:4...	60DE5135	1	Michał	Parus
82	04.11.2017 00:4...	FE9EB2C5	1	Aleksander	Kozioł

Jest tu pole do wpisania liczby n ostatnich wpisów ustawione domyślnie na wartość 100

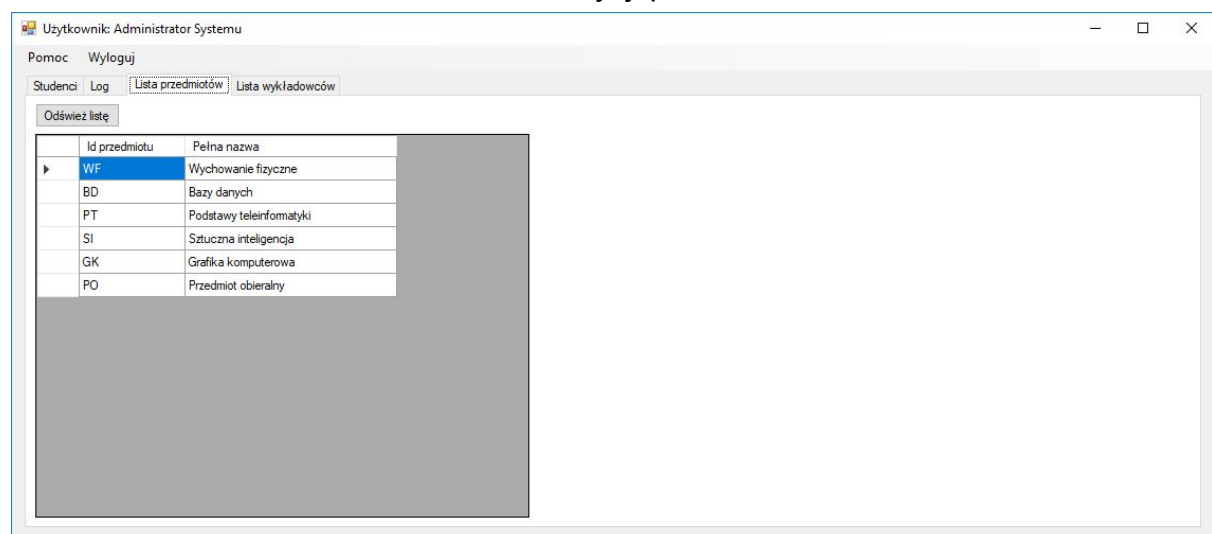
Liczba wierszy:

145 1117 00

Kod odpowiedzialny za wyświetlanie logu:

```
string query = String.Format("SELECT * FROM (SELECT  
a.id,data,a.uid,rdr,b.Imie,b.Nazwisko FROM log a JOIN studenci b on a.UID = b.UID  
ORDER by id DESC LIMIT {0}) sub ORDER by id ASC", n);
```

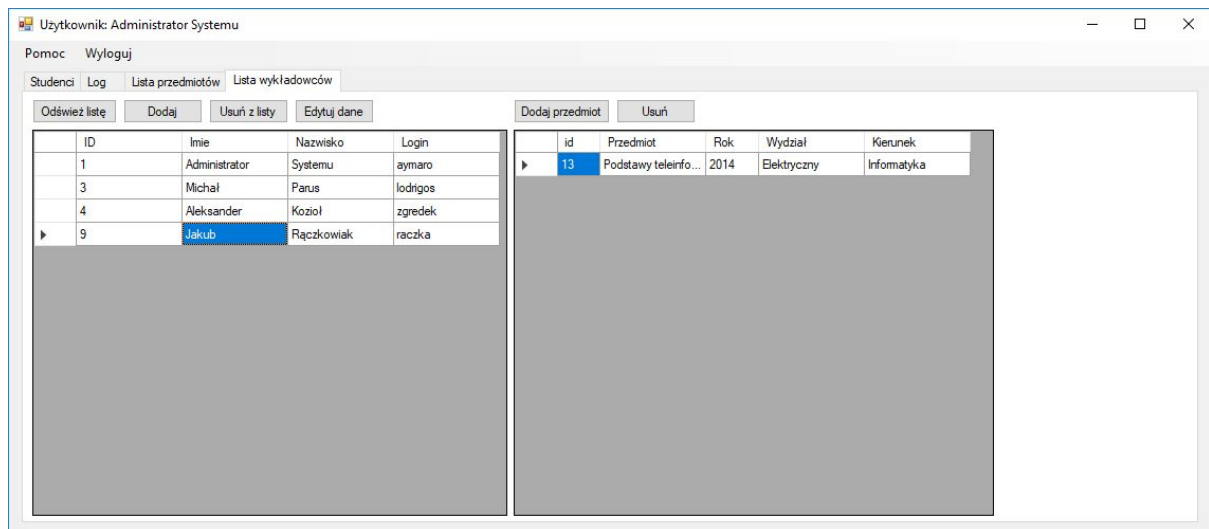
Następną kartą jest widok dostępnych przedmiotów. Jest on niekompletny i brak mu możliwości dodawania, usuwania oraz edycji przedmiotów:



Kod:

```
string query = "SELECT id_przedmiotu, Nazwa FROM przedmioty";
```

Ostatnią zakładką jest widok wykładowców i przedmioty jakie wykładają:



Po wybraniu wykładowcy z listy po lewej, zostanie odświeżona lista jego przedmiotów po prawej stronie z zaznaczeniem dla którego rocznika one są. Kod odpowiedzialny na odświeżanie obu list:

Wykładowcy:

```
string query = "SELECT ID, Imię, Nazwisko, login, hasło FROM wykładowcy";
```

Zajęcia:

```
string query = String.Format("SELECT a.id, c.Nazwa, b.rok, b.wydział, b.kierunek  
FROM wykładowcyZajęcia a JOIN roczniki b on a.rocznik = b.id JOIN przedmioty c  
on a.przedmiot = c.id_przedmiotu WHERE a.wykładowca={0}", id);
```

Wykładowców jak i również przynależne im zajęcia możemy dodawać jak i edytować przez wygodne okna. Okno dodawania wykładowcy wygląda następująco:

Oraz kod odpowiadający za dodanie go do bazy danych:

```
string query = String.Format("INSERT INTO wykładowcy (Imię, Nazwisko, login,  
hasło) VALUES ('{0}', '{1}', '{2}', '{3}'), Imię, Nazwisko,Login,MD5Hash(Hasło));
```

Okno dodawania wykładu dla wybranego wykładowcy z wygodnym systemem podpowiedzi:

Kod:

```
string query = String.Format("INSERT INTO wykadowcyZajecia (wykadowca,
przedmiot, rocznik) values ('{0}', (Select id_przedmiotu from przedmioty where
Nazwa='{1}'), '{2}'))", wykadowcaId, przedmiot, rocznik);
```

Widok dla wykładowców jest dużo prostszy i wygląda on następująco:

Mamy tutaj wybór daty po lewej z pogrubionymi datami oznaczającymi, że w tym dniu mamy jakieś zajęcia. Następnie wyszczególnione zajęcia danego dnia pojawiające się po wybraniu daty oraz lista studentów która powinna znajdować się na konkretnym wykładzie. Kolor wskazuje czy dany student był lub jest obecny na wykładzie z dokładną godziną zgłoszenia obecności. Z tego miejsca wykładowca może ręcznie dodać komuś obecność klikając dwukrotnie na nazwisku studenta. Z tego poziomu wykładowca może również dodać zajęcia przez poniższe okno:

Użytkownik: Jakub Rączkowiak

styczeń 2018						
pon.	wt.	śr.	czw.	pt.	sob.	niedz.
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

Dzisiaj: 28.01.2018

ID: 9 Imię: Jakub Nazwisko: Rączkowiak

Przedmiot: 13/Podstawy teleinformatyki/2014/Elektryczny/Informatyka

Początek zajęć: 8 0 Koniec zajęć: 9 0

Dodaj Anuluj

Lista przedmiotów jest automatycznie uzupełniana dla wygody użytkownika poniższym kodem:

```
string query = String.Format("SELECT a.id, c.Nazwa, b.rok, b.wydział, b.kierunek
FROM wykładowcyZajecia a JOIN roczniki b on a.rocznik = b.id JOIN przedmioty c
on a.przedmiot = c.id_przedmiotu WHERE a.wykladowca={0}", id);
```

Po wprowadzeniu wszystkich danych wykonywane jest poniższe zapytanie:

```
string query = String.Format("INSERT INTO zajecia (wykladowca, przedmiot, data,
początek, koniec, rocznik) VALUES ('{0}', (Select przedmiot from wykładowcyZajecia
where id='{4}'), '{1}', '{2}', '{3}', (Select rocznik from wykładowcyZajecia where
id='{4}'))", wykladowca, data, początek, koniec, id);
```