

**Uniwersytet Śląski
w Katowicach**

Dokumentacja aplikacji
Wirtualny dziekanat

Projekt zespołowy:

Arsen Eloglian

Łukasz Furlepa

Halina Kopicera

Jarosław Kozak

Marek Szafraniec

Katowice 2018

SPIS TREŚCI

I.	ZAŁOŻENIA PROJEKTOWE	3
II.	SPECYFIKACJA WEWNĘTRZNA PROJEKTU	4
2.1.	Model logiczny bazy danych	5
2.2.	Diagram klas	14
III.	SPECYFIKACJA ZEWNĘTRZNA PROJEKTU	37
3.1.	Logowanie do systemu	37
3.2.	Dane osobowe	39
3.3.	Filtrowanie danych	40
3.4.	Wiadomości	41
3.5.	Rola Student	44
3.5.1.	Wyświetlanie przedmiotów	44
3.5.2.	Wnioski	45
3.6.	Rola Nauczyciel	46
3.6.1.	Przypisywanie ocen	46
3.7.	Rola Administrator	47
3.7.1.	Użytkownicy	47
3.7.2.	Dodawanie przedmiotów	48
3.7.3.	Dodawanie wykładowców	48
3.7.4.	Dodawanie studentów	49
3.7.5.	Administrowanie wnioskami	50
	SPIS RYSUNKÓW	52

I. Założenia projektowe

Założenia ogólne

- Witryna internetowa.
- Należy zastosować ORM lub Micro-ORM,
- Dane powinny być walidowane po stronie front i back endu,
- Mechanizm logowania błędów,
- Projekt musi się znajdować na repozytorium,
- Dane muszą być zapisywane i odczytywane z bazy danych,
- Projekt będzie oceniany również ze względu na estetykę,
- Wszystkie zadania muszą być rozpisane w narzędziu do zarządzania projektem ,
- Projekt musi posiadać czytelną dokumentację.

Aplikacja

- Rejestrację nowego użytkownika wraz z podaniem danych osobowych,
- Mechanizm „Przypominania hasła”,
- W systemie muszą występować 3 role – Administrator, Student i Nauczyciel,
- We wszystkich tabelach w systemie powinna być możliwość filtracji danych.

Student

- Wyświetlanie swoich przedmiotów wraz z ocenami,
- Możliwość składania wniosków w postaci elektronicznej (stypendium, przedłużenie sesji, warunek, egzamin komisyjny), po wprowadzeniu szczegółowych danych. Należy jednak umożliwić pobranie całego wniosku w postaci pliku PDF,
- Wyświetlenie historii złożonych wniosków wraz z odpowiednimi statusami (oczekujący, w trakcie rozpatrywania, zaakceptowany, odrzucony),
- Wyświetlanie historii otrzymanych stypendiów.

Nauczyciel

- Lista przedmiotów, która jest przez niego prowadzona,
- Lista studentów, która jest przypisana do danego przedmiotu,
- Wprowadzenie oceny studentowi z danego przedmiotu (I lub II termin).

Administrator

- Wyświetlanie listy aktualnych przedmiotów, studentów i nauczycieli w postaci tabeli,
- Możliwość zablokowania użytkownika lub wymuszenie zmiany hasła,
- Dodawanie i edycja przedmiotów,
- Przypisywanie nauczycieli i studentów do przedmiotów,
- Możliwość wygenerowania raportu ocen końcowych wszystkich studentów wraz ze statystykami (np. ile osób zaliczyło egzamin w I terminie).

II. Specyfikacja wewnętrzna projektu

Aplikacja *Wirtualny dziekanat* powstała środowisku programistycznym Eclipse przy użyciu języka programowania Java. Interfejs graficzny został utworzony poprzez wykorzystanie biblioteki komponentów graficznych GWT. Dane składowane są w bazie Oracle, serwer aplikacji – Wildfly 10.0.

Projekt aplikacji został wykonany w Enterprise Architect. Do organizacji pracy zespołu użyto repozytorium GitHub <https://github.com/szafer/Wirtualny-diekanat>. Zarządzanie pracą zespołu realizowano za pomocą serwisu ASANA <https://app.asana.com/0/459282516818824/list>.

2.1. Model logiczny bazy danych

Tables
Version 1.0
created on 01.11.2017. Last modified 11.01.2018

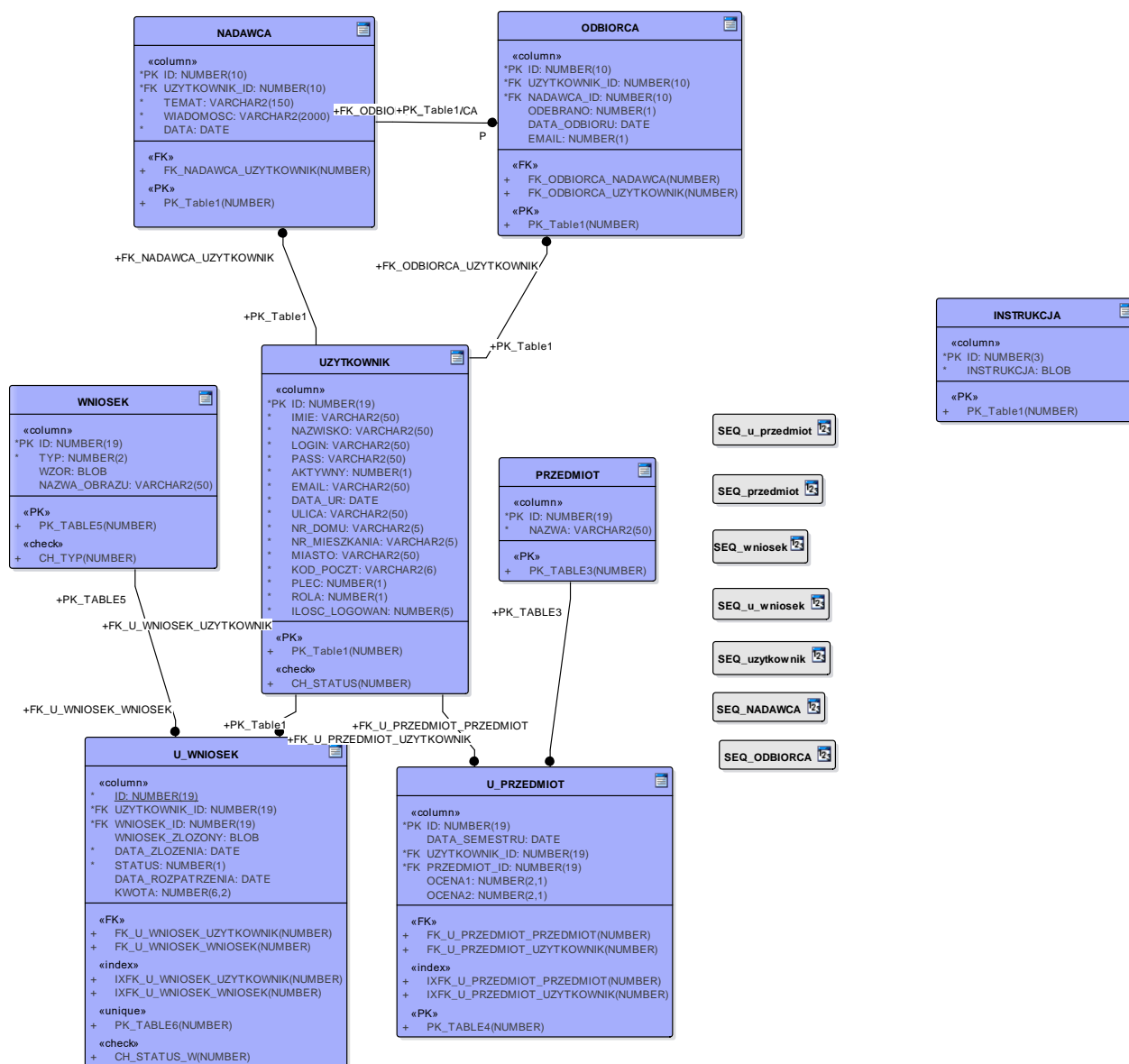


Figure 1: Tables

Database «dbsequence» in package 'Sequences'

```
CREATE SEQUENCE SEQ_nadawca MINVALUE 1 MAXVALUE 99999999999999999999 INCREMENT BY 1 START  
WITH 3000 NOCACHE NOORDER NOCYCLE ;
```

Database «dbsequence» in package 'Sequences'

```
CREATE SEQUENCE SEQ_odbiorca MINVALUE 1 MAXVALUE 99999999999999999999 INCREMENT BY 1
START WITH 3000 NOCACHE NOORDER NOCYCLE ;
```

Database «dbsequence» in package 'Sequences'

```
CREATE SEQUENCE SEQ_przedmiot MINVALUE 1 MAXVALUE 99999999999999999999 INCREMENT BY 1
START WITH 3000 NOCACHE NOORDER NOCYCLE ;
```

Database «dbsequence» in package 'Sequences'

```
CREATE SEQUENCE SEQ_u_przedmiot MINVALUE 1 MAXVALUE 99999999999999999999 INCREMENT BY 1
START WITH 3000 NOCACHE NOORDER NOCYCLE ;
```

Database «dbsequence» in package 'Sequences'

SEQ_u_wniosek
Version 1.0 Phase 1.0 Proposed
created on 09.01.2018. Last modified 09.01.2018
DBMS Oracle

```
CREATE SEQUENCE SEQ_u_wniosek MINVALUE 1 MAXVALUE 99999999999999999999 INCREMENT BY 1
START WITH 3000 NOCACHE NOORDER NOCYCLE ;
```

SEQ_uzytkownik

Database «dbsequence» in package 'Sequences'

SEQ_uzytkownik
Version 1.0 Phase 1.0 Proposed
created on 09.01.2018. Last modified 09.01.2018
DBMS Oracle

```
CREATE SEQUENCE SEQ_uzytkownik MINVALUE 1 MAXVALUE 99999999999999999999 INCREMENT BY 1
START WITH 3000 NOCACHE NOORDER NOCYCLE ;
```

SEQ_wniosek

Database «dbsequence» in package 'Sequences'



SEQ_wniosek
Version 1.0 Phase 1.0 Proposed
created on 09.01.2018. Last modified 09.01.2018
DBMS Oracle


```
CREATE SEQUENCE SEQ_wniosek MINVALUE 1 MAXVALUE 99999999999999999999 INCREMENT BY 1 START
WITH 3000 NOCACHE NOORDER NOCYCLE ;
```

INSTRUKCJA

Tablica z dokumentacją






INSTRUKCJA
Version 1.0 Phase 1.0 Proposed
created on 11.01.2018. Last modified 11.01.2018
DBMS Oracle


COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
 ID	NUMBER(3)	True	
 INSTRUKCJA	BLOB	True	


PRIMARY KEY NAME	COLUMNS	COMMENTS
 PK_Table1	ID	

NADAWCA

NADAWCA
Version 1.0 Phase 1.0 Proposed
created on 11.01.2018. Last modified 11.01.2018
DBMS Oracle

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
 ID	NUMBER(10)	True	
 UZYTKOWNIK_ID	NUMBER(10)	True	
 TEMAT	VARCHAR2(150)	True	
 WIADOMOSC	VARCHAR2(2000)	True	
 DATA	DATE	True	







PRIMARY KEY NAME	COLUMNS	COMMENTS
 PK_Table1	ID	

FOREIGN KEY NAME	COLUMNS	REFERENCES
 FK_NADAWCA_UZYTKOWNIK	UZYTKOWNIK_ID	UZYTKOWNIK(ID)

ODBIORCA



Lista odbiorców wiadomości z potwierdzeniem odebrania

ODBIORCA
Version 1.0 Phase 1.0 Proposed
created on 11.01.2018. Last modified 11.01.2018
DBMS Oracle

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
 ID	NUMBER(10)	True	
 UZYTKOWNIK_ID	NUMBER(10)	True	
 NADAWCA_ID	NUMBER(10)	True	
 ODEBRANO	NUMBER(1)	False	
 DATA_ODBIORU	DATE	False	
 EMAIL	NUMBER(1)	False	



PRIMARY KEY NAME	COLUMNS	COMMENTS
------------------	---------	----------


 PK_Table1	ID	
---	----	--

FOREIGN KEY NAME	COLUMNS	REFERENCES
 FK_ODBIORCA_NADAWCA	NADAWCA_ID	PZI.NADAWCA(ID)
 FK_ODBIORCA_UZYTKOWNIK	UZYTKOWNIK_ID	UZYTKOWNIK(ID)

PRZEDMIOT





PRZEDMIOT
Version 1.0 Phase 1.0 Proposed
created on 01.11.2017. Last modified 01.11.2017
DBMS Oracle








COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
 ID	NUMBER(19)	True	Properties: AutoNum = 1 Increment = 1 property = AutoNum=1;StartNum=1;Increment=1; StartNum = 1
 NAZWA	VARCHAR2(50)	True	


PRIMARY KEY NAME	COLUMNS	COMMENTS
 PK_TABLE3	ID	


UZYTKOWNIK

UZYTKOWNIK
Version 1.0 Phase 1.0 Proposed
created on 01.11.2017. Last modified 12.12.2017
DBMS Oracle

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
 ID	NUMBER(19)	True	Properties: AutoNum = 1 Increment = 1 property = AutoNum=1;StartNum=1;Increment=1; StartNum = 1
 IMIE	VARCHAR2(50)	True	
 NAZWISKO	VARCHAR2(50)	True	
 LOGIN	VARCHAR2(50)	True	







COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
 PASS	VARCHAR2(50)	True	
 AKTYWNY	NUMBER(1)	True	0 - aktywny 1 - zablokowany 2 - wymuszenie zmiany hasła 3 - zalogowany
 EMAIL	VARCHAR2(50)	True	
 DATA_UR	DATE	True	
 ULICA	VARCHAR2(50)	True	
 NR_DOMU	VARCHAR2(5)	True	
 NR_MIESZKANIA	VARCHAR2(5)	True	
 MIASTO	VARCHAR2(50)	True	
 KOD_POCZT	VARCHAR2(6)	True	
 PLEC	NUMBER(1)	True	0 - Mężczyzna 1 - Kobieta
 ROLA	NUMBER(1)	True	0-student 1-wykladowca 2-administrator
 ILOSC_LOGOWAN	NUMBER(5)	True	


PRIMARY KEY NAME	COLUMNS	COMMENTS
 PK_Table1	ID	



TYPE / NAME	COLUMNS	COMMENTS
 «check» CH_STATUS	AKTYWNY	



U_PRZEDMIOT

U_PRZEDMIOT
Version 1.0 Phase 1.0 Proposed
created on 01.11.2017. Last modified 01.11.2017
DBMS Oracle

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
 ID	NUMBER(19)	True	Properties: AutoNum = 1 Increment = 1 property = AutoNum=1;StartNum=1;Increment=1; StartNum = 1
 DATA_SEMESTRU	DATE	False	prezentacja wg algorytmu jeżeli data z pierwszego półrocza prezentacja SEMESTR LETNI $\text{yaer}(\text{data_semestru}) /$ $\text{yaer}(\text{data_semestru}) - 1$ a w przeciwnym razie SEMESTR ZIMOWY $\text{yaer}(\text{data_semestru}) / \text{yaer}(\text{data_semestru}) + 1$
 UZYTKOWNIK_ID	NUMBER(19)	True	
 PRZEDMIOT_ID	NUMBER(19)	True	
 OCENA1	NUMBER(2,1)	False	
 OCENA2	NUMBER(2,1)	False	









PRIMARY KEY NAME	COLUMNS	COMMENTS
 PK_TABLE4	ID	




TYPE / NAME	COLUMNS	COMMENTS
 «index» IXFK_U_PRZEDMIOT_PRZEDMIOT	PRZEDMIOT_ID	
 «index» IXFK_U_PRZEDMIOT_UZYTKOWNIK	UZYTKOWNIK_ID	



FOREIGN KEY NAME	COLUMNS	REFERENCES
 FK_U_PRZEDMIOT_UZYTKOWNIK	UZYTKOWNIK_ID	UZYTKOWNIK(ID)
 FK_U_PRZEDMIOT_PRZEDMIOT	PRZEDMIOT_ID	PRZEDMIOT(ID)

U_WNIOSEK

U_WNIOSEK
Version 1.0 Phase 1.0 Proposed
created on 01.11.2017. Last modified 01.11.2017
DBMS Oracle





COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
 ID	NUMBER(19)	True	Properties: AutoNum = 1 Increment = 1 property = AutoNum=1;StartNum=1;Increment=1; StartNum = 1
 UZYTEKOWNIK_ID	NUMBER(19)	True	
 WNIOSEK_ID	NUMBER(19)	True	
 WNIOSEK_ZLOZONY	BLOB	False	
 DATA_ZLOZENIA	DATE	True	
 STATUS	NUMBER(1)	True	
 DATA_ROZPATRZENIA	DATE	False	0 - oczekujący 1 - w trakcie rozpatrywania 2 - zaakceptowany 3 - odrzucony
 KWOTA	NUMBER(6,2)	False	


TYPE / NAME	COLUMNS	COMMENTS
 «index» IXFK_U_WNIOSEK_UZYTEKOWNIK	UZYTEKOWNIK_ID	
 «index» IXFK_U_WNIOSEK_WNIOSEK	WNIOSEK_ID	
 «check» CH_STATUS_W	STATUS	


FOREIGN KEY NAME	COLUMNS	REFERENCES
 FK_U_WNIOSEK_WNIOSEK	WNIOSEK_ID	WNIOSEK(ID)
 FK_U_WNIOSEK_UZYTEKOWNIK	UZYTEKOWNIK_ID	UZYTEKOWNIK(ID)

WNIOSEK

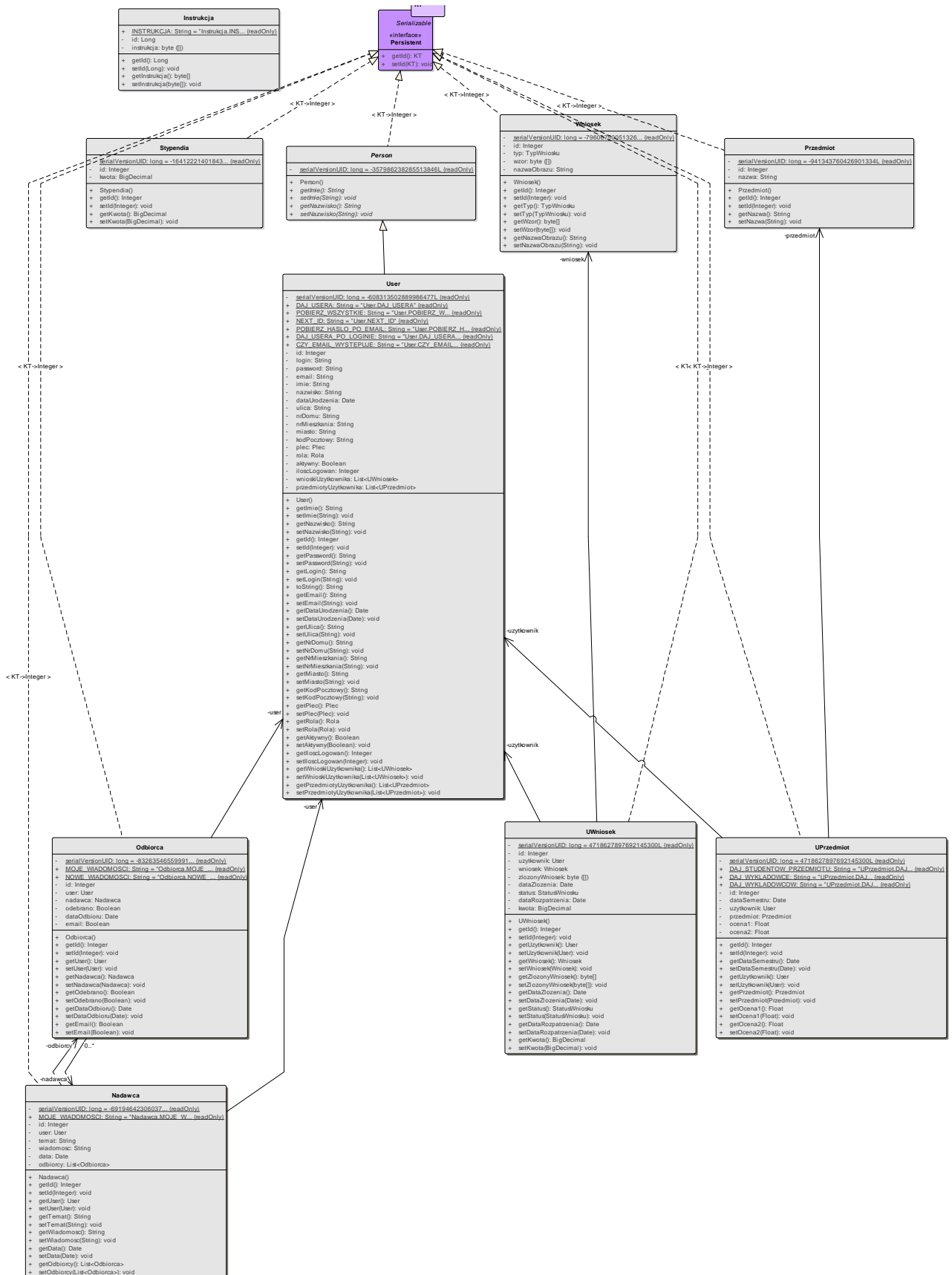
WNIOSEK
Version 1.0 Phase 1.0 Proposed
created on 01.11.2017. Last modified 01.11.2017
DBMS Oracle

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
 ID	NUMBER(19)	True	Properties: AutoNum = 1 Increment = 1 property = AutoNum=1;StartNum=1;Increment=1; StartNum = 1
 TYP	NUMBER(2)	True	0 - stypendium 1 - przedłużenie sesji 2 - warunek 3 - egzamin komisyjny
 WZOR	BLOB	False	
 NAZWA_OBRAZU	VARCHAR2(50)	False	

PRIMARY KEY NAME	COLUMNS	COMMENTS
 PK_TABLE5	ID	




TYPE / NAME	COLUMNS	COMMENTS
 «check» CH_TYP	TYP	





2.2. Diagram klas



Instrukcja

Instrukcja
Version 1.0 Phase 1.0 Proposed
created on 11.01.2018. Last modified 11.01.2018

ATTRIBUTES
 INSTRUKCJA : String Public Const = "Instrukcja.INSTRUKCJA"
 id : Long Private Properties: annotations = @Id
 instrukcja : byte Private Properties: annotations = @Lob

OPERATIONS
 getId () : Long Public
 setId (id : Long) : void Public
 getInstrukcja () : byte Public
 setInstrukcja (instrukcja : byte[]) : void Public

Nadawca

Nadawca
Version 1.0 Phase 1.0 Proposed
created on 11.01.2018. Last modified 11.01.2018

OUTGOING STRUCTURAL RELATIONSHIPS
 Realization from Nadawca to Persistent

ATTRIBUTES
 serialVersionUID : long Private Const = -6919464230603785426L

ATTRIBUTES

MOJE_WIADOMOSCI : String Public Const = "Nadawca.MOJE_WIADOMOSCI"

id : Integer Private

Properties:

annotations = @Id@GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "SEQ_GEN_NADAWCA")@Column(name = "ID")

user : User Private

Properties:

annotations = @ManyToOne(fetch = FetchType.LAZY)@JoinColumn(name = "UZYTKOWNIK_ID")

temat : String Private

Properties:

annotations = @Column(name = "TEMAT", length = 150)

wiadomosc : String Private

Properties:

annotations = @Column(name = "WIADOMOSC", length = 2000)

data : Date Private

Properties:

annotations = @Column(name = "DATA")@Temporal(TemporalType.DATE)

odbiorcy : List<Odbiorca> Private

Properties:

annotations = @OneToMany(fetch = FetchType.LAZY, mappedBy = "nadawca", cascade = CascadeType.PERSIST)

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) Nadawca

Target: Private user (Class) User














Association (direction: Source -> Destination)

Source: Public (Class) Nadawca

Target: Private odbiorcy (Class) Odbiorca
Cardinality: [0..*]

Association (direction: Source -> Destination)

ASSOCIATIONS	
Source: Public (Class) Odbiorca	Target: Private nadawca (Class) Nadawca

OPERATIONS
 Nadawca () : Public
 getId () : Integer Public Properties: annotations = @Override
 setId (id : Integer) : void Public Properties: annotations = @Override
 getUser () : User Public
 setUser (user : User) : void Public
 getTemat () : String Public
 setTemat (temat : String) : void Public
 getWiadomosc () : String Public
 setWiadomosc (wiadomosc : String) : void Public
 getData () : Date Public
 setData (data : Date) : void Public
 getOdbiorcy () : List<Odbiorca> Public
 setOdbiorcy (odbiory : List<Odbiorca>) : void Public

OUTGOING STRUCTURAL RELATIONSHIPS

← Realization from Odbiorca to Persistent

ATTRIBUTES

serialVersionUID : long Private Const = -8326354655999127073L

MOJE_WIADOMOSCI : String Public Const = "Odbiorca.MOJE_WIADOMOSCI"

NOWE_WIADOMOSCI : String Public Const = "Odbiorca.NOWE_WIADOMOSCI"

id : Integer Private

Properties:

annotations = @Id@GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "SEQ_GEN_ODBIORCA")@Column(name = "ID")

user : User Private

Properties:

annotations = @ManyToOne(fetch = FetchType.LAZY)@JoinColumn(name = "UZYTKOWNIK_ID")

nadawca : Nadawca Private

Properties:

annotations = @ManyToOne(fetch = FetchType.LAZY)@JoinColumn(name = "NADAWCA_ID")

odebrano : Boolean Private

Properties:

annotations = @Column(name = "ODEBRANO")

dataOdbioru : Date Private

Properties:

annotations = @Column(name = "DATA_ODBIORU")@Temporal(TemporalType.DATE)

email : Boolean Private

Properties:

ATTRIBUTES

annotations = @Column(name = "EMAIL")

ASSOCIATIONS

 Association (direction: Source -> Destination)

Source: Public (Class) Odbiorca

Target: Private nadawca (Class) Nadawca

 Association (direction: Source -> Destination)

Source: Public (Class) Odbiorca


Target: Private user (Class) User


 Association (direction: Source -> Destination)

Source: Public (Class) Nadawca

Target: Private odbiorcy (Class) Odbiorca
Cardinality: [0..*]

OPERATIONS


 Odbiorca () : Public


 getId () : Integer Public


Properties:
annotations = @Override


 setId (id : Integer) : void Public


Properties:
annotations = @Override






 getUser () : User Public

 setUser (user : User) : void Public

 getNadawca () : Nadawca Public

 setNadawca (nadawca : Nadawca) : void Public


 getOdebrano () : Boolean Public


OPERATIONS
 setOdebrano (odebrano : Boolean) : void Public
 getDataOdbioru () : Date Public
 setDataOdbioru (dataOdbioru : Date) : void Public
 getEmail () : Boolean Public
 setEmail (email : Boolean) : void Public



Person

Person
Version 1.0 Phase 1.0 Proposed
created on 11.01.2018. Last modified 11.01.2018

OUTGOING STRUCTURAL RELATIONSHIPS
 Realization from Person to Persistent

INCOMING STRUCTURAL RELATIONSHIPS
 Generalization from User to Person

ATTRIBUTES
 serialVersionUID : long Private Const = -357986238285513846L

OPERATIONS
 Person () : Public
 getImie () : String Public

OPERATIONS

- ◆ setImie (imie : String) : void Public
- ◆ getNazwisko () : String Public
- ◆ setNazwisko (nazwisko : String) : void Public

Przedmiot

Przedmiot
Version 1.0 Phase 1.0 Proposed
created on 11.01.2018. Last modified 11.01.2018

OUTGOING STRUCTURAL RELATIONSHIPS

- ⬅ Realization from Przedmiot to Persistent

ATTRIBUTES

- ◆ serialVersionUID : long Private Const = -941343760426901334L
- ◆ id : Integer Private
Properties:
annotations = @Id@GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "SEQ_GEN_PRZEDMIOT")@Column(name = "ID")
- ◆ nazwa : String Private
Properties:
annotations = @Column(name = "NAZWA")

ASSOCIATIONS

- ✎ Association (direction: Source -> Destination)
Source: Public (Class) UPrzedmiot
Target: Private przedmiot (Class) Przedmiot

OPERATIONS

- ◆ Przedmiot () : Public

OPERATIONS

◆ getId () : Integer Public

Properties:
annotations = @Override

◆ setId (id : Integer) : void Public

Properties:
annotations = @Override

◆ getNazwa () : String Public

◆ setNazwa (nazwa : String) : void Public

Stypendia

Stypendia
Version 1.0 Phase 1.0 Proposed
created on 11.01.2018. Last modified 11.01.2018

OUTGOING STRUCTURAL RELATIONSHIPS

↩ Realization from Stypendia to Persistent

ATTRIBUTES

◆ serialVersionUID : long Private Const = -1641222140184318577L

◆ id : Integer Private

Properties:
annotations = @Id@Column(name = "ID")

◆ kwota : BigDecimal Private

@JoinColumn(name = "U_WNIOSEK_ID") private UWniosek wniosekUzytkownika;

Properties:
annotations = @Column(name = "KWOTA")

OPERATIONS

◆ Stypendia () : Public

OPERATIONS
<p>💎 getId () : Integer Public</p> <p>Properties: annotations = @Override</p>
<p>💎 setId (id : Integer) : void Public</p> <p>Properties: annotations = @Override</p>
<p>💎 getKwota () : BigDecimal Public</p> <p>public UWniosek getWniosekUzytkownika() { return wniosekUzytkownika; } public void setWniosekUzytkownika(UWniosek wniosekUzytkownika) { this.wniosekUzytkownika = wniosekUzytkownika; }</p>
<p>💎 setKwota (kwota : BigDecimal) : void Public</p>

UPrzedmiot

UPrzedmiot
Version 1.0 Phase 1.0 Proposed
created on 11.01.2018. Last modified 11.01.2018

OUTGOING STRUCTURAL RELATIONSHIPS
<p>➡ Realization from UPrzedmiot to Persistent</p>

ATTRIBUTES
<p>💎 serialVersionUID : long Private Const = 4718627897692145300L</p>
<p>💎 DAJ_STUDENTOW_PRZEDMIOTU : String Public Const = "UPrzedmiot.DAJ_STUDENTOW_PRZEDMIOTU"</p>
<p>💎 DAJ_WYKLADOWCE : String Public Const = "UPrzedmiot.DAJ_WYKLADOWCE"</p>
<p>💎 DAJ_WYKLADOWCOW : String Public Const = "UPrzedmiot.DAJ_WYKLADOWCOW"</p>
<p>💎 id : Integer Private</p> <p>Properties:</p>

ATTRIBUTES

annotations = @Id@GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "SEQ_GEN_U_PRZEDMIOT")@Column(name = "ID")

dataSemestru : Date Private

Properties:

annotations = @Column(name = "DATA_SEMESTRU")@Temporal(TemporalType.DATE)

uzytkownik : User Private

Properties:

annotations = @ManyToOne(fetch = FetchType.LAZY)@JoinColumn(name = "UZYTKOWNIK_ID")

przedmiot : Przedmiot Private

Properties:

annotations = @ManyToOne(fetch = FetchType.LAZY)@JoinColumn(name = "PRZEDMIOT_ID")

ocena1 : Float Private

Properties:

annotations = @Column(name = "OCENA1")

ocena2 : Float Private

Properties:

annotations = @Column(name = "OCENA2")

ASSOCIATIONS

 Association (direction: Source -> Destination)

Source: Public (Class) UPrzedmiot


Target: Private uzytkownik (Class) User

 Association (direction: Source -> Destination)

Source: Public (Class) UPrzedmiot


Target: Private przedmiot (Class) Przedmiot

OPERATIONS

 getId () : Integer Public



Properties:

annotations = @Override


OPERATIONS
 setId (id : Integer) : void Public Properties: annotations = @Override
 getDataSemestru () : Date Public
 setDataSemestru (dataSemestru : Date) : void Public
 getUzytkownik () : User Public
 setUzytkownik (uzytkownik : User) : void Public
 getPrzedmiot () : Przedmiot Public
 setPrzedmiot (przedmiot : Przedmiot) : void Public
 getOcena1 () : Float Public
 setOcena1 (ocena1 : Float) : void Public
 getOcena2 () : Float Public
 setOcena2 (ocena2 : Float) : void Public













UWniosek

UWniosek
Version 1.0 Phase 1.0 Proposed
created on 11.01.2018. Last modified 11.01.2018

OUTGOING STRUCTURAL RELATIONSHIPS
 Realization from UWniosek to Persistent
ATTRIBUTES
 serialVersionUID : long Private Const = 4718627897692145300L

ATTRIBUTES	
 id : Integer Private	Properties: annotations = @Id@GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "SEQ_GEN_U_WNIOSEK")@Column(name = "ID")
 uzytkownik : User Private	Properties: annotations = @ManyToOne(fetch = FetchType.LAZY)@JoinColumn(name = "UZYTKOWNIK_ID")
 wniosek : Wniosek Private	Properties: annotations = @ManyToOne(fetch = FetchType.LAZY)@JoinColumn(name = "WNIOSEK_ID")
 zlozonyWniosek : byte Private	Properties: annotations = @Lob@Column(name = "WNIOSEK_ZLOZONY")
 dataZlozenia : Date Private	Properties: annotations = @Column(name = "DATA_ZLOZENIA")@Temporal(TemporalType.DATE)
 status : StatusWniosku Private	Properties: annotations = @Column(name = "STATUS")@Enumerated(EnumType.ORDINAL)
 dataRozpatrzenia : Date Private	Properties: annotations = @Column(name = "DATA_ROZPATRZENIA")@Temporal(TemporalType.DATE)
 kwota : BigDecimal Private	Properties: annotations = @Column(name = "KWOTA")
ASSOCIATIONS	
 Association (direction: Source -> Destination)	

ASSOCIATIONS	
Source: Public (Class) UWniosek	Target: Private wniosek (Class) Wniosek
 Association (direction: Source -> Destination)	
Source: Public (Class) UWniosek	Target: Private uzytkownik (Class) User

OPERATIONS
 UWniosek () : Public
 getId () : Integer Public Properties: annotations = @Override
 setId (id : Integer) : void Public Properties: annotations = @Override
 getUzytkownik () : User Public
 setUser (uzytkownik : User) : void Public
 getWniosek () : Wniosek Public
 setWniosek (wniosek : Wniosek) : void Public
 getZlozonyWniosek () : byte Public
 setZlozonyWniosek (zlozonyWniosek : byte[]) : void Public
 getDataZlozenia () : Date Public
 setDataZlozenia (dataZlozenia : Date) : void Public
 getStatus () : StatusWniosku Public

OPERATIONS
◆ setStatus (status : StatusWniosku) : void Public
◆ getDataRozpatrzenia () : Date Public
◆ setDataRozpatrzenia (dataRozpatrzenia : Date) : void Public
◆ getKwota () : BigDecimal Public
◆ setKwota (kwota : BigDecimal) : void Public

User

User
Version 1.0 Phase 1.0 Proposed
created on 11.01.2018. Last modified 11.01.2018
Extends Person

OUTGOING STRUCTURAL RELATIONSHIPS
↩ Generalization from User to Person

ATTRIBUTES
◆ serialVersionUID : long Private Const = -608313502889986477L
◆ DAJ_USERA : String Public Const = "User.DAJ_USERA"
◆ POBIERZ_WSZYSTKIE : String Public Const = "User.POBIERZ_WSZYSTKIE"
◆ NEXT_ID : String Public Const = "User.NEXT_ID"
◆ POBIERZ_HASLO_PO_EMAIL : String Public Const = "User.POBIERZ_HASLO_PO_EMAIL"
◆ DAJ_USERA_PO_LOGINIE : String Public Const = "User.DAJ_USERA_PO_LOGINIE"

ATTRIBUTES

🔹 `CZY_EMAIL_WYSTĘPUJE : String Public Const = "User.CZY_EMAIL_WYSTĘPUJE"`

🔹 `id : Integer Private`

`public static final String ID = "id"; public static final String IMIE = "imie"; public static final String NAZWISKO = "nazwisko";`

Properties:

`annotations = @Id@GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "SEQ_GEN_USER")@Column(name = "ID")`

🔹 `login : String Private`

Dane logowania

Properties:

`annotations = @Column(name = "LOGIN")`

🔹 `password : String Private`

Properties:

`annotations = @Column(name = "PASS")`

🔹 `email : String Private`

Properties:

`annotations = @Column(name = "EMAIL")`

🔹 `imie : String Private`

Dane osobowe

Properties:

`annotations = @Column(name = "IMIE")`

🔹 `nazwisko : String Private`

Properties:

`annotations = @Column(name = "NAZWISKO")`

🔹 `dataUrodzenia : Date Private`


Properties:

`annotations = @Column(name = "DATA_UR")@Temporal(TemporalType.DATE)`

🔹 `ulica : String Private`

ATTRIBUTES	
	Properties: annotations = @Column(name = "ULICA")
 nrDomu : String Private	Properties: annotations = @Column(name = "NR_DOMU")
 nrMieszkania : String Private	Properties: annotations = @Column(name = "NR_MIESZKANIA")
 miasto : String Private	Properties: annotations = @Column(name = "MIASTO")
 kodPocztowy : String Private	Properties: annotations = @Column(name = "KOD_POCZT")
 plec : Plec Private	Properties: annotations = @Column(name = "PLEC")@Enumerated(EnumType.ORDINAL)
 rola : Rola Private	Properties: annotations = @Column(name = "ROLA")@Enumerated(EnumType.ORDINAL)
 aktywny : Boolean Private	Properties: annotations = @Column(name = "AKTYWNY")
 iloscLogowan : Integer Private licznik w dÅŁ, ile logowaŁ, do zmiany hasŁ,a	Properties: annotations = @Column(name = "ILOSC_LOGOWAN")


ATTRIBUTES

 wnioskiUzytkownika : List<UWniosek> Private

```
@ManyToMany(fetch = FetchType.LAZY, cascade = CascadeType.ALL)
@JoinTable(name = "U_WNIOSEK", joinColumns = { @JoinColumn(name = "UZYTKOWNIK_ID", nullable = false,
updatable = false) }, inverseJoinColumns = {
@JoinColumn(name = "WNIOSEK_ID", nullable = false, updatable = false) }) private List<Wniosek> wnioski;
```

Properties:

annotations = @OneToMany(fetch = FetchType.LAZY, mappedBy = "uzytkownik", cascade = CascadeType.ALL)

 przedmiotyUzytkownika : List<UPzedmiot> Private

Properties:

annotations = @OneToMany(fetch = FetchType.LAZY, mappedBy = "uzytkownik", cascade = CascadeType.ALL)

ASSOCIATIONS

 Association (direction: Source -> Destination)

Source: Public (Class) Nadawca

Target: Private user (Class) User

 Association (direction: Source -> Destination)

Source: Public (Class) Odbiorca

Target: Private user (Class) User

 Association (direction: Source -> Destination)

Source: Public (Class) UPrzedmiot


Target: Private uzytkownik (Class) User


 Association (direction: Source -> Destination)


Source: Public (Class) UWniosek


Target: Private uzytkownik (Class) User

OPERATIONS

 User () : Public

 getImie () : String Public

 setImie (imie : String) : void Public

 getNazwisko () : String Public

OPERATIONS

◆ setNazwisko (nazwisko : String) : void Public

◆ getId () : Integer Public

Properties:
annotations = @Override

◆ setId (id : Integer) : void Public

Properties:
annotations = @Override

◆ getPassword () : String Public

◆ setPassword (password : String) : void Public

◆ getLogin () : String Public

◆ setLogin (login : String) : void Public

◆ toString () : String Public

◆ getEmail () : String Public

◆ setEmail (email : String) : void Public

◆ getDataUrodzenia () : Date Public

◆ setDataUrodzenia (dataUrodzenia : Date) : void Public

◆ getUlica () : String Public

◆ setUlica (ulica : String) : void Public

◆ getNrDomu () : String Public

OPERATIONS
◆ setNrDomu (nrDomu : String) : void Public
◆ getNrMieszkania () : String Public
◆ setNrMieszkania (nrMieszkania : String) : void Public
◆ getMiasto () : String Public
◆ setMiasto (miasto : String) : void Public
◆ getKodPocztowy () : String Public
◆ setKodPocztowy (kodPocztowy : String) : void Public
◆ getPlec () : Plec Public
◆ setPlec (plec : Plec) : void Public
◆ getRola () : Rola Public
◆ setRola (rola : Rola) : void Public
◆ getAktywny () : Boolean Public
◆ setAktywny (aktywny : Boolean) : void Public
◆ getIloscLogowan () : Integer Public
◆ setIloscLogowan (iloscLogowan : Integer) : void Public
◆ getWnioskiUzytkownika () : List<UWniosek> Public

OPERATIONS

- ◆ setWnioskiUzytkownika (wnioskiUzytkownika : List<UWniosek>) : void Public
- ◆ getPrzedmiotyUzytkownika () : List<UPrzedmiot> Public
- ◆ setPrzedmiotyUzytkownika (przedmiotyUzytkownika : List<UPrzedmiot>) : void Public

Wniosek

Wniosek
Version 1.0 Phase 1.0 Proposed
created on 11.01.2018. Last modified 11.01.2018

OUTGOING STRUCTURAL RELATIONSHIPS

- ↩ Realization from Wniosek to Persistent

ATTRIBUTES

- ◆ serialVersionUID : long Private Const = -7960675005132642988L
- ◆ id : Integer Private
Properties:
 annotations = @Id@Column(name = "ID")
- ◆ typ : TypWniosku Private
Properties:
 annotations = @Column(name = "TYP")@Enumerated(EnumType.ORDINAL)
- ◆ wzor : byte Private
Properties:
 annotations = @Lob@Basic(fetch = FetchType.LAZY)@Column(name = "WZOR")
- ◆ nazwaObrazu : String Private
Properties:
 annotations = @Column(name = "NAZWA_OBRAZU")


ASSOCIATIONS

 Association (direction: Source -> Destination)


Source: Public (Class) UWniosek

Target: Private wniosek (Class) Wniosek


OPERATIONS

 Wniosek () : Public

@OneToMany(fetch = FetchType.LAZY, mappedBy = "wniosek") private List<UWniosek> wnioskiUzytkownika;

 getId () : Integer Public


Properties:
annotations = @Override


 setId (id : Integer) : void Public


Properties:
annotations = @Override

 getTyp () : TypWniosku Public

 setTyp (typ : TypWniosku) : void Public

 getWzor () : byte Public

 setWzor (wzor : byte[]) : void Public

 getNazwaObrazu () : String Public

 setNazwaObrazu (nazwaObrazu : String) : void Public

Persistent

Interface in package 'System'

Persistent
Version 1.0 Phase 1.0 Proposed
created on 11.01.2018. Last modified 11.01.2018
Extends Serializable

INCOMING STRUCTURAL RELATIONSHIPS
⇒ Realization from Wniosek to Persistent
⇒ Realization from Nadawca to Persistent
⇒ Realization from Stypendia to Persistent
⇒ Realization from Odbiorca to Persistent
⇒ Realization from UWniosek to Persistent
⇒ Realization from Person to Persistent
⇒ Realization from UPrzedmiot to Persistent
⇒ Realization from Przedmiot to Persistent
OPERATIONS
◆ getId () : KT Public
◆ setId (id : KT) : void Public

III. Specyfikacja zewnętrzna projektu

Przeznaczeniem aplikacji *Wirtualny dziekanat* jest udostępnienie studentom oraz wykładowcom możliwości rejestracji zdarzeń zachodzących podczas roku akademickiego. Z uwagi na różnorodność zagadnień realizowanych poprzez wymienione grupy użytkowników, uprawnienia do wykonywania poszczególnych operacji uzależnione zostały od roli przypisanej danemu użytkownikowi. W systemie wyszczególniono następujące role:

- student,
- wykładowca,
- administrator.

3.1. Logowanie do systemu

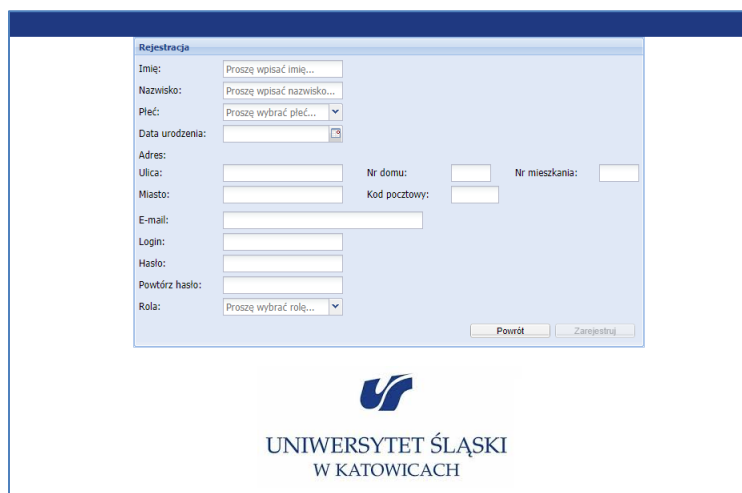
Pierwszą formatką aplikacji jest formatka logowania. Na formatce należy podać nazwę użytkownika i hasło. Formatka została uzupełniona o opcję przypominania hasła: po jej wyborze użytkownik otrzyma maila z wygenerowanym tymczasowym hasłem.



The screenshot shows a web application interface for 'Wirtualny dziekanat'. At the top, there is a dark blue header bar with the text 'Nie masz konta?' and a link 'Zarejestruj się'. Below the header, the title 'Wirtualny dziekanat' is centered. On the left side, there is a login form with the following elements: a label 'Wpisz login:' followed by a text input field containing 'a1'; a label 'Wpisz hasło:' followed by a password input field containing '*'; a link 'Przypomnienie hasła' in blue text; and a 'Loguj' button. In the center of the page, there is a logo of the University of Silesia in Katowice, consisting of a stylized 'U' and 'S' in blue, with the text 'UNIwersytet śląski W KATOWICACH' below it.

Rys. 1. Logowanie do systemu

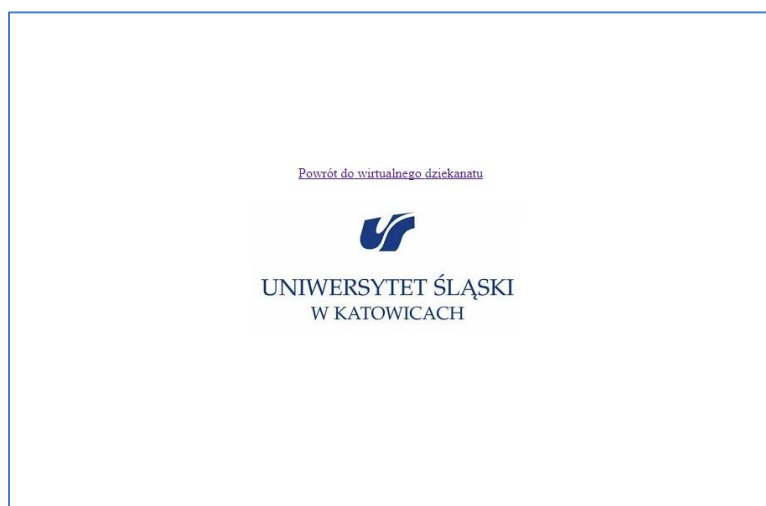
Użytkownik niezarejestrowany ma możliwość dokonania rejestracji poprzez wybór opcji ***Zarejestruj się.***



The image shows a web registration form titled "Rejestracja" (Registration) for the University of Silesia in Katowice. The form is contained within a light blue box with a dark blue header. It includes the following fields: "Imię:" (First name) with a text input and a placeholder "Proszę wpisać imię..."; "Nazwisko:" (Surname) with a text input and a placeholder "Proszę wpisać nazwisko..."; "Płeć:" (Gender) with a dropdown menu and a placeholder "Proszę wybrać płeć..."; "Data urodzenia:" (Date of birth) with a date picker; "Adres:" (Address) section with "Ulica:" (Street) and "Nr domu:" (House number) inputs; "Miasto:" (City) and "Kod pocztowy:" (Postal code) inputs; "E-mail:" (Email) with a text input; "Login:" (Login) with a text input; "Hasło:" (Password) with a text input; "Powtórz hasło:" (Repeat password) with a text input; and "Rola:" (Role) with a dropdown menu and a placeholder "Proszę wybrać rolę...". At the bottom right of the form are two buttons: "Powrót" (Back) and "Zarejestruj" (Register). Below the form, centered, is the logo of the University of Silesia in Katowice, consisting of a stylized blue 'U' and 'S' monogram, followed by the text "UNIWERSYTET ŚLĄSKI W KATOWICACH".

Rys. 2. Rejestracja użytkownika

Po zakończeniu pracy (systemowe zamknięcie), użytkownikowi zostaje wyświetlona formatka umożliwiająca powrót do ponownego logowania.



Rys. 3. Powrót do logowania

3.2. Dane osobowe

Każdy zalogowany użytkownik posiada dostęp do swoich danych osobowych, które są dla niego dostępne do edycji. Funkcjonalność dostępna po wyborze opcji *Dane > Moje dane*. Pozycją niedostępną dla użytkownika jest przypisana mu *Rola* – jedynie administrator systemu może przypisać użytkownikowi inną *Rolę*.

The screenshot shows a web interface for editing personal data. The top navigation bar includes links: **Dane osobowe**, **Przedmioty**, **Wiadomości**, **Druki i wnioski**, **Instrukcja**, and **Zamknij**. Below this is a sub-header **Moje dane**. The form fields are as follows:

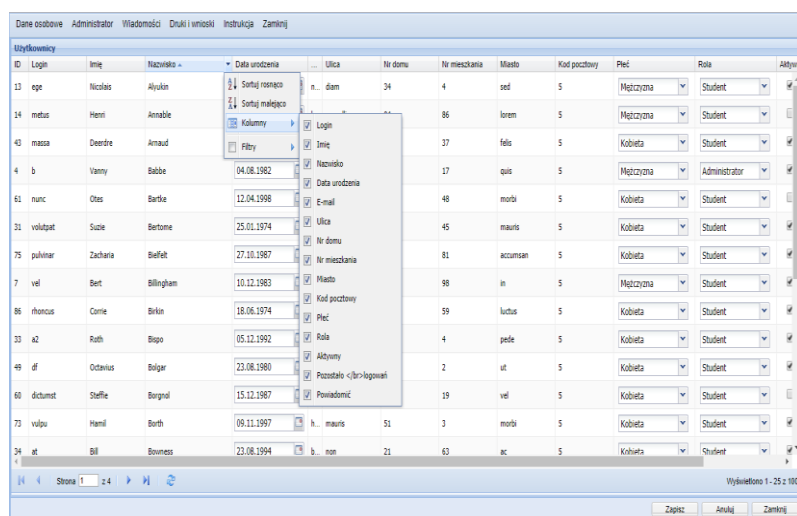
Label	Value
Imię:	Cloris
Nazwisko:	Kenrack
Płeć:	Mężczyzna
Data urodzenia:	09.01.1991
Adres:	
Ulica:	rutrum
Nr domu:	62
Nr mieszkania:	18
Miasto:	morbi
Kod pocztowy:	5
E-mail:	ckenrackj@java.com
Login:	a0
Hasło:	••
Powtórz hasło:	••
Rola:	Student

At the bottom right of the form are three buttons: **Zapisz**, **Anuluj**, and **Zamknij**.

Rys. 4. Dane osobowe

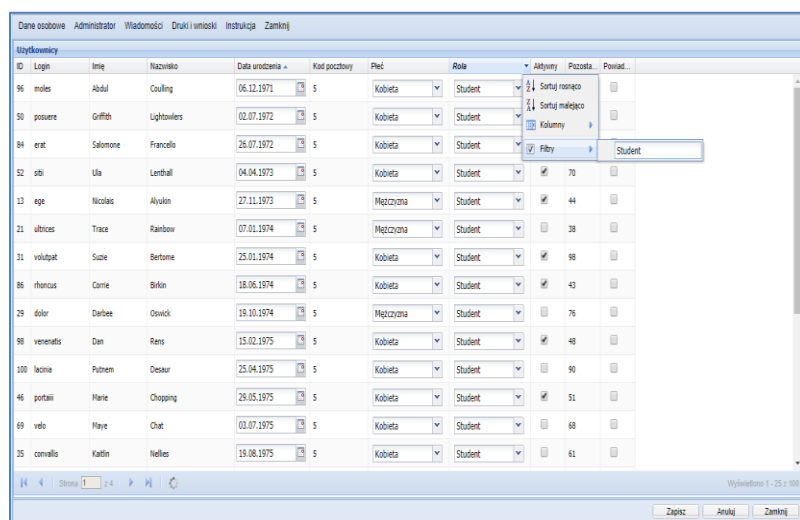
3.3. Filtrowanie danych

Zasadą zastosowaną w całej aplikacji jest możliwość filtrowania danych prezentowanych w postaci tabelarycznej. Filtrowania można dokonać poprzez ograniczenie kolumn prezentowanych w tabeli. W tym celu, należy *kliknąć* w prawy brzeg dowolnej kolumny, co spowoduje wyświetlenie menu podręcznego, zaprezentowanego na Rys. 5. Z listy kolumn należy odznaczyć kolumny, których nie chcemy obserwować w trakcie dalszej pracy.



Rys. 5. Menu podręczne filtrowania kolumn

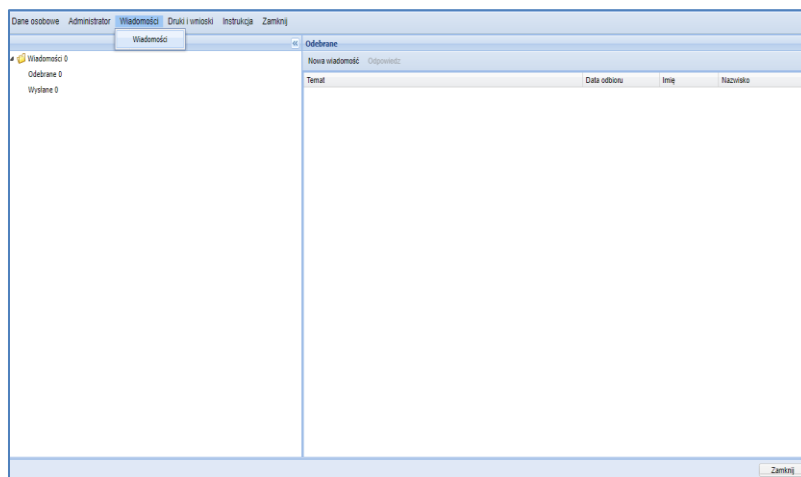
Drugim zakresem udostępnionym do filtrowania jest ilość wierszy, którą można ograniczyć poprzez filtrowanie dowolnej z kolumn, w zależności od wartości w niej występujących. Przykład ograniczenia zakresu danych zaprezentowano na Rys. 6.



Rys. 6. Menu podręczne filtrowania zakresu danych

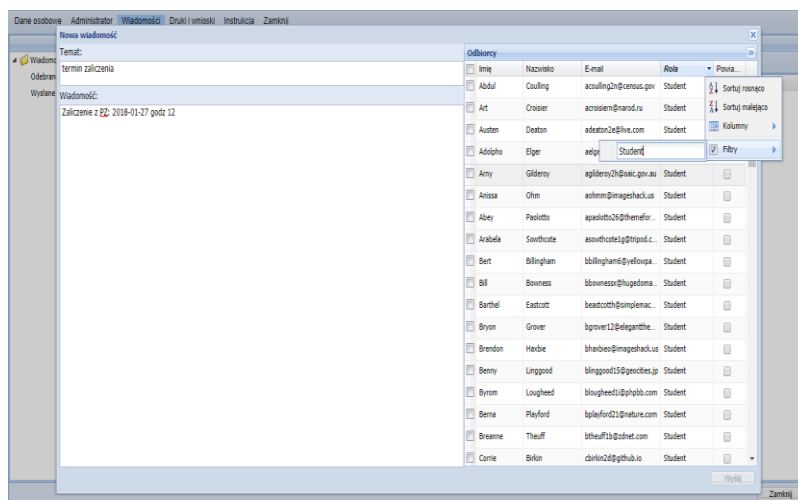
3.4. Wiadomości

Każdy zalogowany użytkownik posiada dostęp do modułu Wiadomości (opcja menu *Wiadomości*). Moduł ten służy do komunikowania się między sobą, użytkowników systemu. Dane prezentowane są w postaci drzewka folderów, z podziałem na wiadomości odebrane i wysłane. Każda z pozycji prezentuje również licznik wiadomości.



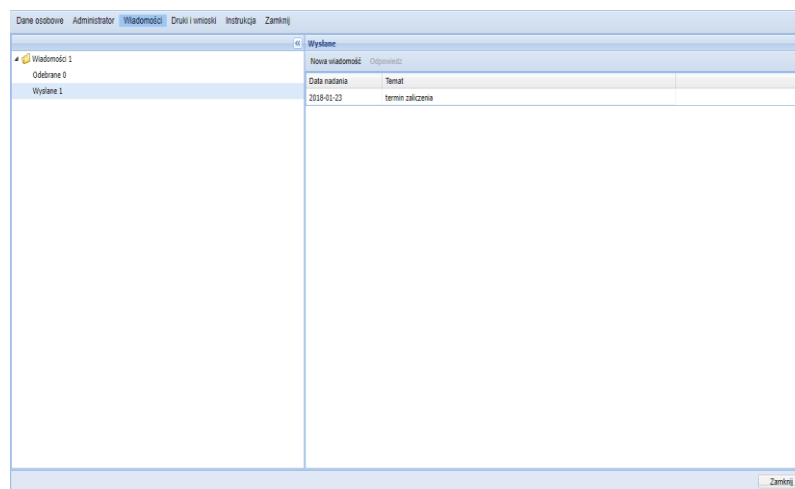
Rys. 7. Wiadomości

Po wyborze opcji *Nowa wiadomość*, zostanie otwarte okno wiadomości przedstawione na Rys. 8. W oknie tym należy wpisać tytuł oraz treść wiadomości, a następnie wybrać użytkowników do których wiadomość zostanie wysłana. Użytkowników wybiera się z listy poprzez zaznaczenie pola odbiorcy. Dodatkowo można ustawić wysyłanie powiadomień systemowych, poprzez zaznaczenia pola *Powiadomienie*.



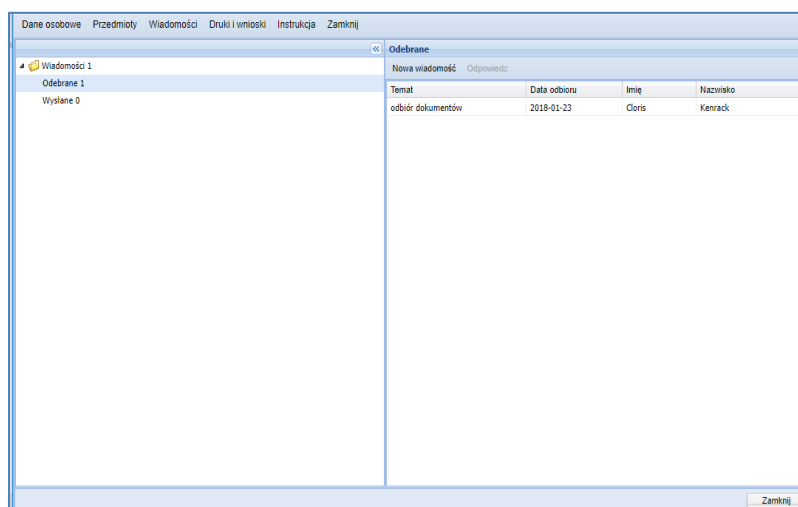
Rys. 8. Nowa wiadomość

Po wysłaniu wiadomości następuje uaktualnienie folderów, a w oknie po prawej stronie zostaje wyświetlona lista wiadomości wysłanych.



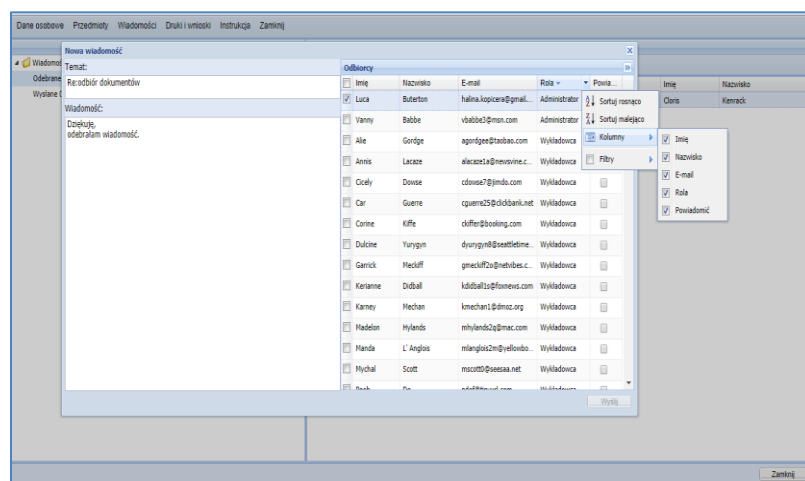
Rys. 9. Wiadomości wysłane

Analogicznie, po zaznaczeniu pozycji *Odebrane*, w okienku po prawej stronie ekranu zostanie wyświetlona lista odebranych wiadomości, co przedstawiono na Rys.10.



Rys. 10. Wiadomości odebrane

Aplikacja umożliwia szybką odpowiedź na odebraną wiadomość, poprzez użycie opcji *Odpowiedź*. Analogicznie jak podczas wysyłania nowej wiadomości, udostępnione zostaje okno do wyboru użytkowników, do których wiadomość zostanie wysłana.



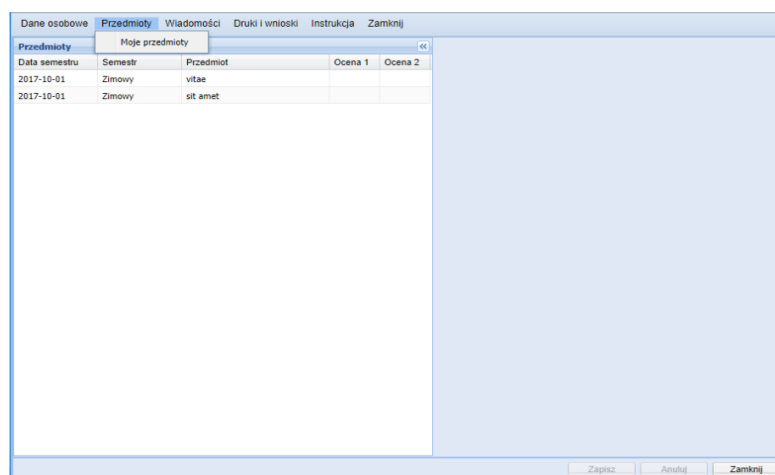
Rys. 11. Odpowiadanie na wiadomość

3.5. Rola Student

Użytkownik z przypisaną rolą *Student*, w *Wirtualnym dziekanacie* ma możliwość przeglądania oraz zmiany swoich danych osobowych, przeglądania wybranych przez siebie przedmiotów, składania przewidzianych systemowo wniosków, wyświetlania historii złożonych przez siebie wniosków oraz otrzymanych stypendiów.

3.5.1. Wyświetlanie przedmiotów

Każdy student ma możliwość wyświetlania listy przedmiotów na które uczęszcza, wraz z ocenami uzyskanymi na zakończenie semestru. Funkcjonalność też udostępniono po wyborze opcji *Przedmiot > Moje przedmioty*.



Data semestru	Semestr	Przedmiot	Ocena 1	Ocena 2
2017-10-01	Zimowy	vitae		
2017-10-01	Zimowy	sit amet		

Rys. 12. Student – przedmioty

3.5.2. Wnioski

Opcja Druki i wnioski jest kolejną pozycją menu, dostępną dla każdego zalogowanego użytkownika. Wybierając z menu pozycję Druki i wnioski, student ma możliwość złożenia wniosku stypendialnego, wniosku o przedłużenie sesji, warunkowego zaliczenia sesji czy też egzamin komisyjny.

The screenshot shows a web application interface for submitting requests. The main menu at the top includes 'Dane osobowe', 'Przedmioty', 'Wiadomości', 'Druki i wnioski', 'Instrukcja', and 'Zamknij'. The 'Druki i wnioski' tab is selected. Below the tabs, there's a 'Dodaj wniosek' section. It features a date input field set to '12.01.2018', a dropdown menu for 'Typ wniosku' with a list of options including 'Egzamin komisyjny', 'Przedłużenie sesji', 'Warunek', and 'Egzamin komisyjny', and a 'Status wniosku' field set to 'Oczekujący'. There are 'Anuluj' and 'Zapisz' buttons. To the right, there's a 'Drukuj' button and an error message: 'Error Typ wniosku: To pole jest wymagane'. At the bottom of the form, there are 'Zapisz', 'Anuluj', and 'Zamknij' buttons.

Rys. 13. Student wnioski

3.6. Rola Nauczyciel

Użytkownik z przypisaną rolą *Nauczyciel*, w *Wirtualnym dziekanacie* ma możliwość przeglądania oraz zmiany swoich danych osobowych, przeglądania prowadzonych przedmiotów wraz z listą studentów uczęszczających na zajęcia z tych przedmiotów. Moduł umożliwia również wpisywanie ocen.

3.6.1. Przypisywanie ocen

Każdy wykładowca ma możliwość wyświetlania listy przedmiotów przez siebie prowadzonych w poszczególnych semestrach. Funkcjonalność tę udostępniono po wyborze opcji *Przedmiot > Przypisywanie ocen*.

Po zaznaczeniu wybranego przedmiotu, w okienku po prawej stronie ekranu, zostanie wyświetlona lista studentów uczęszczających na dany przedmiot. Lista umożliwia bezpośrednie wpisywanie ocen – I i II termin.

Przedmiot			Oceny studentów			
Data semestru	Semestr	Przedmiot	Imię	Nazwisko	Ocena 1	Ocena 2
2017-10-01	Zimowy	aenean	Bert	Billingham	5	
2017-10-01	Zimowy	quis orci	Lisle	Scholz	4.5	
2017-10-01	Zimowy	curae nulla dapibus	Price	Steart		
2017-10-01	Zimowy	sem				
2017-10-01	Zimowy	suscipit a				
2017-10-01	Zimowy	convallis				
2017-10-01	Zimowy	at nunc				
2017-10-01	Zimowy	eu sapien				
2017-10-01	Zimowy	curae dui faucibus				
2017-10-01	Zimowy	phasellus in				
2017-10-01	Zimowy	sit amet eleifend				

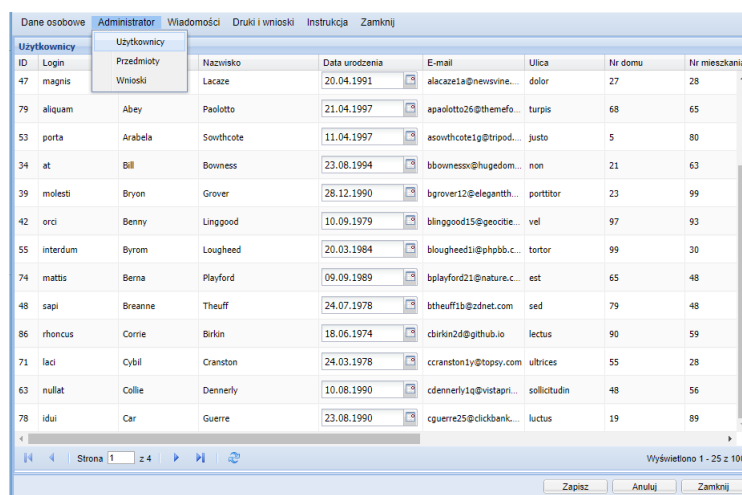
Rys. 14. Nauczyciel - przedmioty

3.7. Rola Administrator

Użytkownik z przypisaną rolą *Administrator*, w *Wirtualnym dziekanacie* ma możliwość wyświetlania listy aktualnych przedmiotów, studentów i nauczycieli w postaci tabeli, możliwość zablokowania użytkownika lub wymuszenie zmiany hasła, dodawania i edycji przedmiotów jak również przypisywania nauczycieli i studentów do przedmiotów. Dodatkową funkcjonalnością jest możliwość wygenerowania raportu ocen końcowych wszystkich studentów wraz ze statystykami (np. ile osób zaliczyło egzamin w I terminie).

3.7.1. Użytkownicy

Administrator ma możliwość wyświetlania listy wszystkich użytkowników w formie tabeli. Funkcjonalność tę udostępniono po wyborze opcji *Administrator > Użytkownicy*. Każda kolumna tabeli umożliwia filtrowanie jej zawartości. Administrator ma prawo do zmiany danych osobowych (za wyjątkiem hasła użytkownika).



Dane osobowe Administrator Wiadomości Druki i wnioski Instrukcja Zamknij									
Użytkownicy									
ID	Login	Przedmioty	Wnioski	Nazwisko	Data urodzenia	E-mail	Ulica	Nr domu	Nr mieszkania
47	magnis			Lacaze	20.04.1991	alacaze1a@newsvine...	dolor	27	28
79	aliquam	Abey		Paolotto	21.04.1997	apaolotto26@themefo...	turpis	68	65
53	porta	Arabela		Southcote	11.04.1997	esouthcote1g@tripod...	justo	5	80
34	at	Bill		Bowness	23.08.1994	bbownessx@hugedom...	non	21	63
39	molesti	Bryon		Grover	28.12.1990	bgrover12@eleganth...	porttitor	23	99
42	orci	Benny		Linggood	10.09.1979	blinggood15@geotie...	vel	97	93
55	interdum	Byrom		Lougheed	20.03.1984	blougheed1i@phbb.c...	tortor	99	30
74	mattis	Berna		Playford	09.09.1989	bplayford21@nature.c...	est	65	48
48	sapi	Breanne		Theuff	24.07.1978	bttheuff1b@zdnnet.com	sed	79	48
86	rhoncus	Corrie		Birkin	18.06.1974	cbirkin2d@github.io	lectus	90	59
71	iaci	Cybil		Cranston	24.03.1978	ccranston1y@topsy.com	ultrices	55	28
63	nullat	Collie		Dennerly	10.08.1990	cdennerly1q@vistapri...	sollicitudin	48	56
78	idui	Car		Guerre	23.08.1990	cguerre25@clickbank...	luctus	19	89

Rys. 15. Administrator - użytkownicy

Do funkcji administratora należy potwierdzanie rejestracji użytkownika poprzez zaznaczenie pozycji *Aktywny*, może on również zablokować użytkownika, odznaczając pozycję *Aktywny*

Nr domu	Nr mieszkania	Miasto	Kod pocztowy	Płeć	Rola	Aktywny	Pozosta...	Powiad...
57	24	adipiscing	5	Kobieta	Student	<input type="checkbox"/>	38	<input type="checkbox"/>
29	27	eget	5	Męczyzna	Student	<input checked="" type="checkbox"/>	19	<input type="checkbox"/>
81	43	nulla	5	Kobieta	Wykładowca	<input checked="" type="checkbox"/>	50	<input type="checkbox"/>
95	93	fusce	5	Kobieta	Student	<input type="checkbox"/>	55	<input type="checkbox"/>
92	19	nulla	5	Kobieta	Student	<input type="checkbox"/>	38	<input type="checkbox"/>
12	70	amet	5	Męczyzna	Wykładowca	<input checked="" type="checkbox"/>	90	<input type="checkbox"/>
27	28	eget	5	Kobieta	Wykładowca	<input checked="" type="checkbox"/>	8	<input type="checkbox"/>
13	60	elit	5	Męczyzna	Student	<input type="checkbox"/>	28	<input type="checkbox"/>
68	65	fusce	5	Kobieta	Student	<input type="checkbox"/>	86	<input type="checkbox"/>
5	80	magna	5	Kobieta	Student	<input checked="" type="checkbox"/>	24	<input type="checkbox"/>
43	98	in	5	Męczyzna	Student	<input checked="" type="checkbox"/>	86	<input type="checkbox"/>
21	63	ac	5	Kobieta	Student	<input checked="" type="checkbox"/>	75	<input type="checkbox"/>
65	69	lorem	5	Męczyzna	Student	<input type="checkbox"/>	69	<input type="checkbox"/>

Rys. 16. Administrator- edycja użytkowników

3.7.2. Dodawanie przedmiotów

Opcja menu *Administrator > Przedmioty* udostępnia listę przedmiotów wraz z prowadzącymi je wykładowcami oraz uczęszczającymi na nie studentami. Kliknięcie na klawisz *Dodaj przedmiot*, wprowadza na listę pusty wiersz w którym należy wpisać nazwę przedmiotu.

Nazwa	Data semestru	Semestr	Wykładowca
Nowy przedmiot			

Rys. 17. Administrator – dodawanie przedmiotu

3.7.3. Dodawanie wykładowców

Opcja menu *Administrator > Przedmioty* udostępnia listę przedmiotów wraz z prowadzącymi je wykładowcami oraz uczęszczającymi na nie studentami. Kliknięcie na klawisz *Dodaj*

wykładowcę, wprowadza na listę pusty wiersz w którym należy wypełnić datę semestru oraz wybrać wykładowcę z listy rozwijanej, utworzonej na podstawie wszystkich zarejestrowanych w systemie wykładowców.

Rys. 18. Administrator - dodawanie wykładowcy

3.7.4. Dodawanie studentów

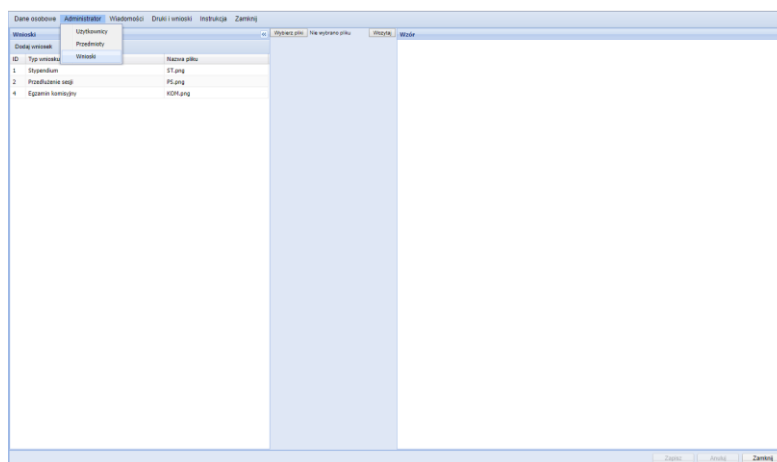
Opcja menu *Administrator > Przedmioty* udostępnia listę przedmiotów wraz z prowadzącymi je wykładowcami oraz uczęszczającymi na nie studentami. Kliknięcie na klawisz *Dodaj studenta*, wprowadza na listę pusty wiersz w którym należy wypełnić datę semestru oraz wybrać studenta z listy rozwijanej, utworzonej na podstawie wszystkich zarejestrowanych w systemie studentów.

Rys. 19. Administrator - dodawanie studentów

3.7.5. Administrowanie wnioskami

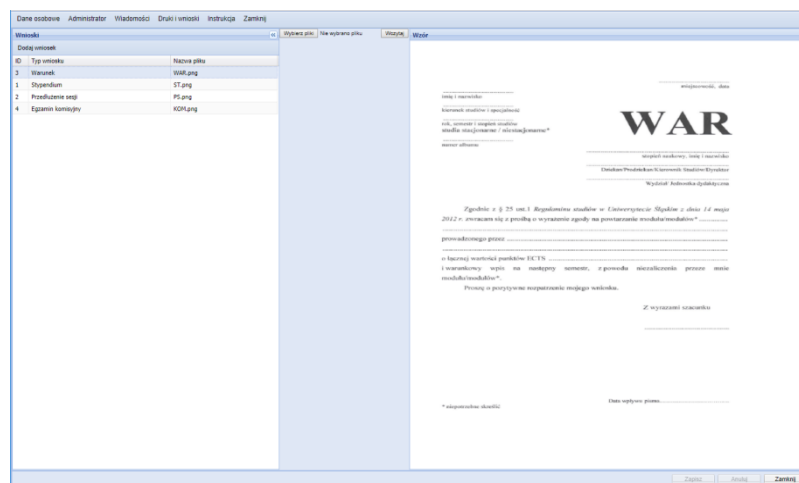
Opcja *Administrator > Wnioski* umożliwia zarządzanie, czyli wprowadzanie, usuwanie oraz zmianę nazwy wniosków, udostępnionych do rejestracji w systemie.

Po wyborze opcji, na formatce zostanie wyświetlona lista wniosków.



Rys. 20. Administrowanie wnioskami

Zaznaczenie pozycji w liście wniosków, spowoduje wyświetlenie podglądu danego wniosku, co zostało zaprezentowane na Rys.21



Rys. 21. Podgląd wzoru wniosku

Kliknięcie na klawisz *Dodaj wniosek*, wprowadza na listę pusty wiersz w którym należy wpisać nazwę wniosku.

SPIS RYSUNKÓW

RYS. 1. LOGOWANIE DO SYSTEMU	37
RYS. 2. REJESTRACJA UŻYTKOWNIKA	38
RYS. 3. POWRÓT DO LOGOWANIA	38
RYS. 4. DANE OSOBOWE	39
RYS. 5. MENU PODRĘCZNE FILTROWANIA KOLUMN	40
RYS. 6. MENU PODRĘCZNE FILTROWANIA ZAKRESU DANYCH	40
RYS. 7. WIADOMOŚCI	41
RYS. 8. NOWA WIADOMOŚĆ	41
RYS. 9. WIADOMOŚCI WYŚLANE	42
RYS. 10. WIADOMOŚCI ODEBRANE	42
RYS. 11. ODPOWIADANIE NA WIADOMOŚĆ	43
RYS. 12. STUDENT - PRZEDMIOTY	44
RYS. 13. STUDENT WNIOSKI	45
RYS. 14. NAUCZYCIEL - PRZEDMIOTY	46
RYS. 15. ADMINISTRATOR - UŻYTKOWNICY	47
RYS. 16. ADMINISTRATOR- EDYCJA UŻYTKOWNIKÓW	48
RYS. 17. ADMINISTRATOR – DODAWANIE PRZEDMIOTU	48
RYS. 18. ADMINISTRATOR - DODAWANIE WYKŁADOWCY	49
RYS. 19. ADMINISTRATOR - DODAWANIE STUDENTÓW	49
RYS. 20. ADMINISTROWANIE WNIOSKAMI	50
RYS. 21. PODGLĄD WZORU WNIOSKU	50
RYS. 22. ADMINISTRATOR - DODAWANIE WNIOSKU	51
RYS. 23. ADMINISTRATOR - WYBÓR WZORU WNIOSKU	51
RYS. 24. ADMINISTRATOR - WCZYTANIE WZORU WNIOSKU	51