

# ***Koncepcja projektowa – Bazy Danych 2***

## ***System rekrutacji na studia***

### ***Zespół 15***

Jakub Kordel  
Łukasz Reszka  
Jakub Strawa  
Jarosław Zabuski

## ***Spis treści:***

OPIS SYSTEMU .....	3
CO CHCEMY POKRYĆ BAZĄ I APLIKACJĄ? .....	3
PROPONOWANY SPOSÓB REALIZACJI .....	3
PROPONOWANE TERMINY REALIZACJI ETAPÓW .....	3
SŁOWNIK DZIEDZINY .....	3
USE CASE'Y - JAROSŁAW ZABUSKI: .....	4
OPIS PRZYPADKÓW UŻYCIA: .....	9
SCENARIUSZE WYKONANIA WYBRANYCH PRZYPADKÓW UŻYCIA .....	11
DIAGRAMY KLAS: .....	15
DIAGRAMY SEKWENCJI: .....	17
MACIERZ CRUD - ŁUKASZ RESZKA: .....	20
MODEL E-R - JAKUB KORDEL, JAKUB STRAWA: .....	21
MODEL LOGICZNY - JAKUB KORDEL: .....	23
MODEL FIZYCZNY – JAKUB KORDEL: .....	25
STORYBOARD'Y – JAKUB STRAWA: .....	29
WYMAGANIA NIEFUNKCJONALNE – JAKUB KORDEL: .....	31

## Opis systemu

System ma na celu realizację rekrutacji na studia licencjackie/inżynierskie konkretnej uczelni wyższej. System jest dostępny dla pracowników uprawnionych do przyjmowania studentów oraz kandydatów, którzy wpisują swoje dane do formularza oraz preferencje wyboru z dostępnych kierunków i wydziałów. System przygotowuje listy osób przyjętych na podstawie dostępnych miejsc na kierunkach, zasad wydziałów oraz preferencji wyboru kandydatów.

## Co chcemy pokryć bazą i aplikacją?

Wprowadzanie danych przez kandydatów (między innymi punktów z egzaminu maturalnego), dodawanie wydziałów i kierunków wraz z liczbami miejsc przez osobę odpowiedzialną za rekrutację. W podanym terminie, system tworzy listy osób przyjętych oraz listy rezerwowe według wprowadzonego algorytmu. System dopuszcza posiadanie przez studenta pierwszeństwa na poszczególne lub wszystkie kierunki (np. laureaci olimpiad). Tak wygenerowane listy system udostępnia w formie tabeli (np. w Excelu, PDF'ie). Kandydat może wyrazić brak chęci uczestniczenia w dalszych rekrutacjach, jeśli dostał się na kierunek z wyboru innego niż pierwszy lub nie dostał się w ogóle. Przed drugą turą system umożliwia osobom odpowiedzialnym za rekrutację wprowadzenie informacji o tym czy dany kandydat złożył dokumenty. Następnie w ustalonym terminie system generuje nowe listy odrzucając z poprzednich osoby, które nie złożyły dokumentów.

## Proponowany sposób realizacji

Baza danych: SQLite 3

Aplikacja: Java, środowisko IntelliJ

## Proponowane terminy realizacji etapów

- Zdefiniowanie modelu pojęciowego (26.11)
- Zdefiniowanie logicznego modelu danych dla bazy relacyjnej oraz projekt aplikacji (17.12)
- Stworzenie modelu fizycznego, implementacja bazy danych i aplikacji, zapewnienie danych testowych, testy (04.02, ewentualnie oddanie tydzień wcześniej)

## Słownik dziedziny

**Kandydat (syn. Aplikant)** – klient systemu oraz jego główny użytkownik. Jednym z głównych celów systemu jest zapewnienie mu komfortowej obsługi jego aplikacji na studia na docelowej uczelni wyższej.

**Kierunek studiów** – główny cel aplikacji składanej przez kandydata na studia. Wybierany przez aplikanta i oficjalnie przypisywany mu przez system po zamknięciu danej tury rekrutacyjnej. Zakładamy, że w jednej aplikacji na studia dany kandydat może wybrać do pięciu, uszeregowanych według priorytetów kierunków studiów. System ma za cel przypisać aplikanta do najwyższego możliwego do spełnienia priorytetu i zmieszczeniu się w limicie studentów na dany kierunek.

**Priorytet** – w kontekście systemu istotność, z jaką traktujemy dany wybór kandydata na studia co do kierunku, jaki chce studiować. Kandydat ma w aplikacji wybrać, jakie kierunki chce studiować oraz

z jakim priorytetem, od 1 (najbardziej pożądanego kierunku studiów) do 5 (najmniej pożądanego kierunku studiów). Nie jest konieczne wykorzystanie całej skali.

**Aplikacja** – oficjalna prośba o przyjęcie na studia, składana przez kandydata do systemu uczelnianego. Zawiera ona dane osobowe pewnego kandydata na obsługiwanej uczelni wyższej, listę kierunków posortowaną wobec priorytetów, a także dowód przesłania opłaty rekrutacyjnej.

**Pracownik** – podtyp użytkownika systemu, który mając dostęp do uzyskanych przez system list aplikantów na studia dalej je przetwarza. Rozróżniamy pracowników na pracowników dziekanatu, którzy utworzone listy aplikantów wykorzystują do przyjmowania poszczególnych kandydatów na studia, oraz administratorów systemu, którzy jako jedyni mają prawo do zmieniania danych wprowadzonych przez kandydatów, w ramach korekcji błędów zgłoszonych przez aplikantów.

**Rekrutacja** – proces polegający na wygenerowaniu w zadanym przedziale czasowym list osób przyjętych na wszystkie kierunki na wszystkich wydziałach na danej uczelni. Dopuszczamy przeprowadzenie kilku tur w ramach jednej rekrutacji, gdzie każdy aplikant może wyrazić chęć lub niechęć do brania w niej udziału. Ważne jest jednak to, że aplikant, jeśli znajduje się już na liście przyjętych na dany kierunek nie może zostać przeniesiony na kierunek o niższym priorytecie. Aplikanci, którzy dostali się na kierunek najwyższego priorytetu automatycznie nie biorą udziału w dalszych turach rekrutacji.

**Punktacja** – liczba punktów uzyskanych przez aplikanta w rekrutacji na studia. Jest to zazwyczaj liczba punktów uzyskanych na maturze lub jej zagranicznym odpowiedniku znormalizowana według określonego wzoru. Aplikant, który wykazał się ponad przeciętnymi wynikami w nauce (finalista olimpiady przedmiotowej, laureat indeksu uczelnianego) automatycznie dostaje maksymalną liczbę punktów podczas rekrutacji na wszystkie lub tylko niektóre kierunki.

**Administrator systemu** – pracownik uczelni odpowiedzialny za utrzymanie bazy danych oraz systemu informatycznego odpowiedzialnego za rekrutację. Posiada on możliwość bezpośredniej modyfikacji i wprowadzania danych do systemu, jeśli zajdzie taka potrzeba.

**Pracownik dziekanatu** – pracownik wydziału uczelni odpowiedzialny za wprowadzanie danych do systemów wewnątrz-wydziałowych oraz przeprowadzenie części procesu rekrutacyjnego, tj. zbieranie dokumentów potwierdzających osiągnięte wyniki w nauce, zdjęcia, potwierdzenia zapłaty opłat.

Use Case'y - Jarosław Zabuski:

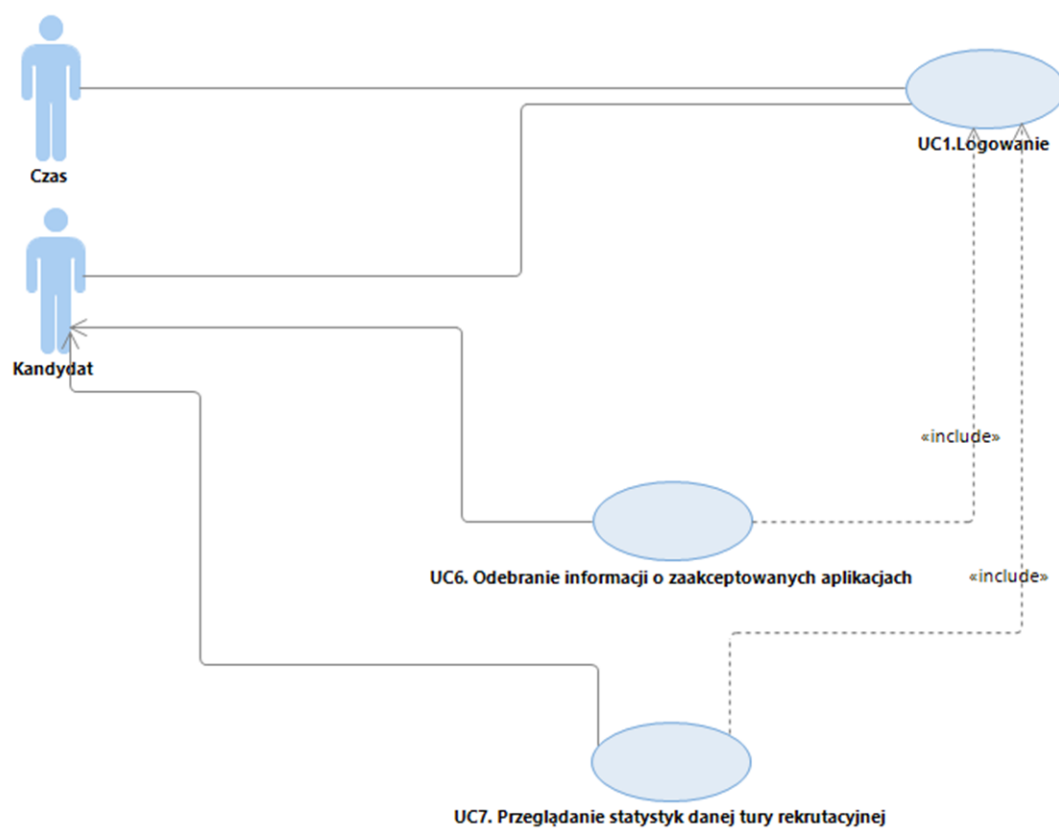


Diagram 1: Diagram przypadków użycia, przedstawiający akcje możliwe do wykonania przez Kandydata po zamknięciu tury rekrutacji, w której brał udział.

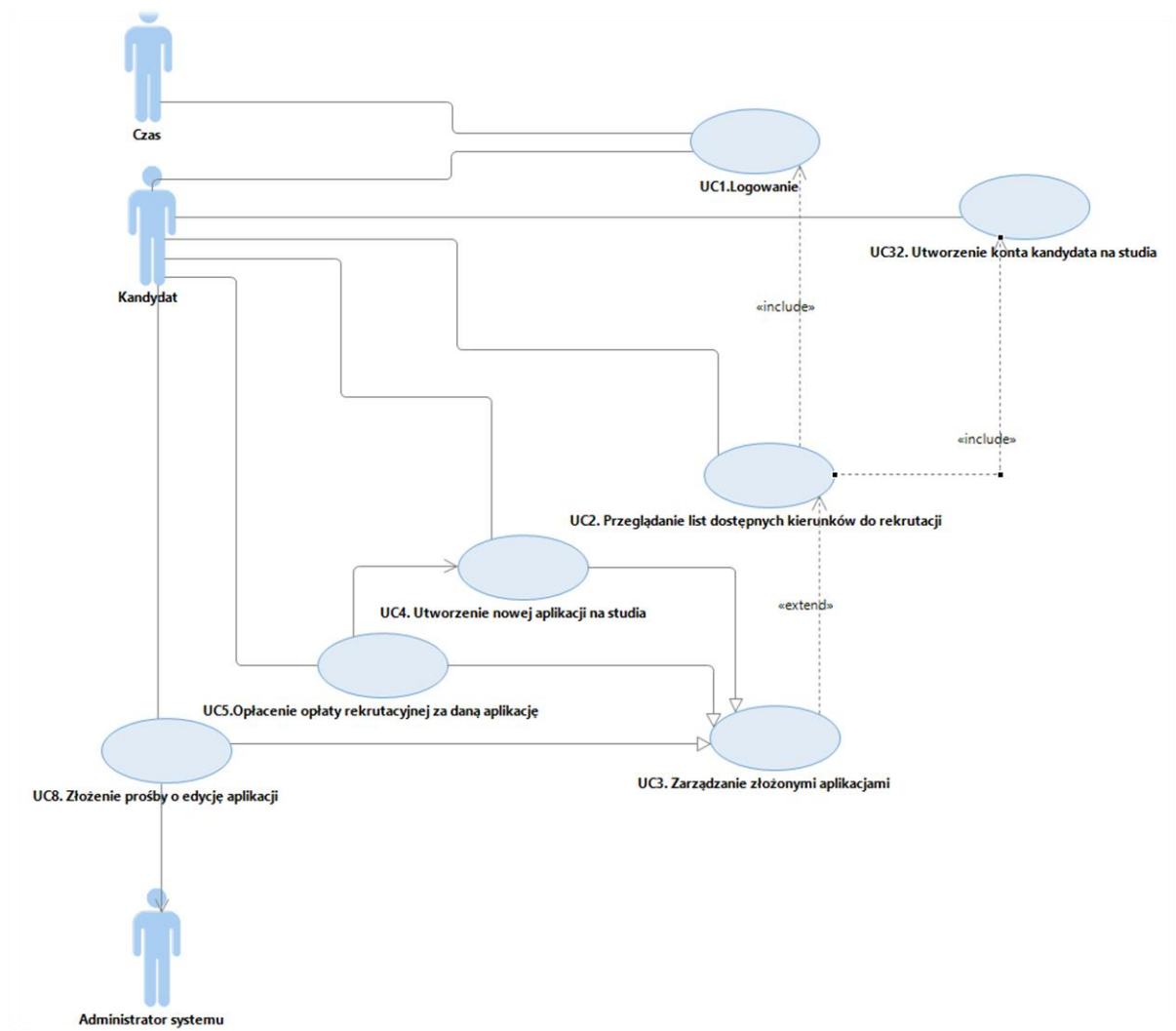


Diagram 2: Diagram przypadków użycia, przedstawiający akcje możliwe do wykonania przez Kandydata w trakcie tury rekrutacji, w której uczestnictwo zadeklarował.

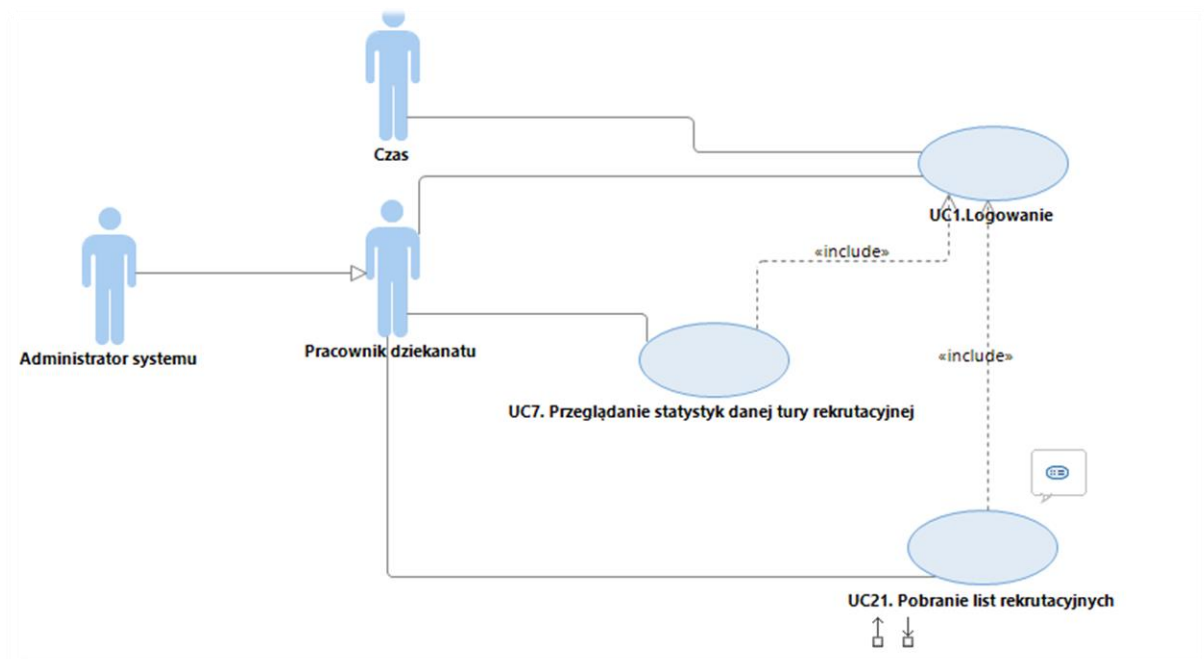


Diagram 3: Diagram przypadków użycia, przedstawiający akcje możliwe do wykonania przez Pracownika po upływie terminu zakończenia obecnie modyfikowanej tury rekrutacji.

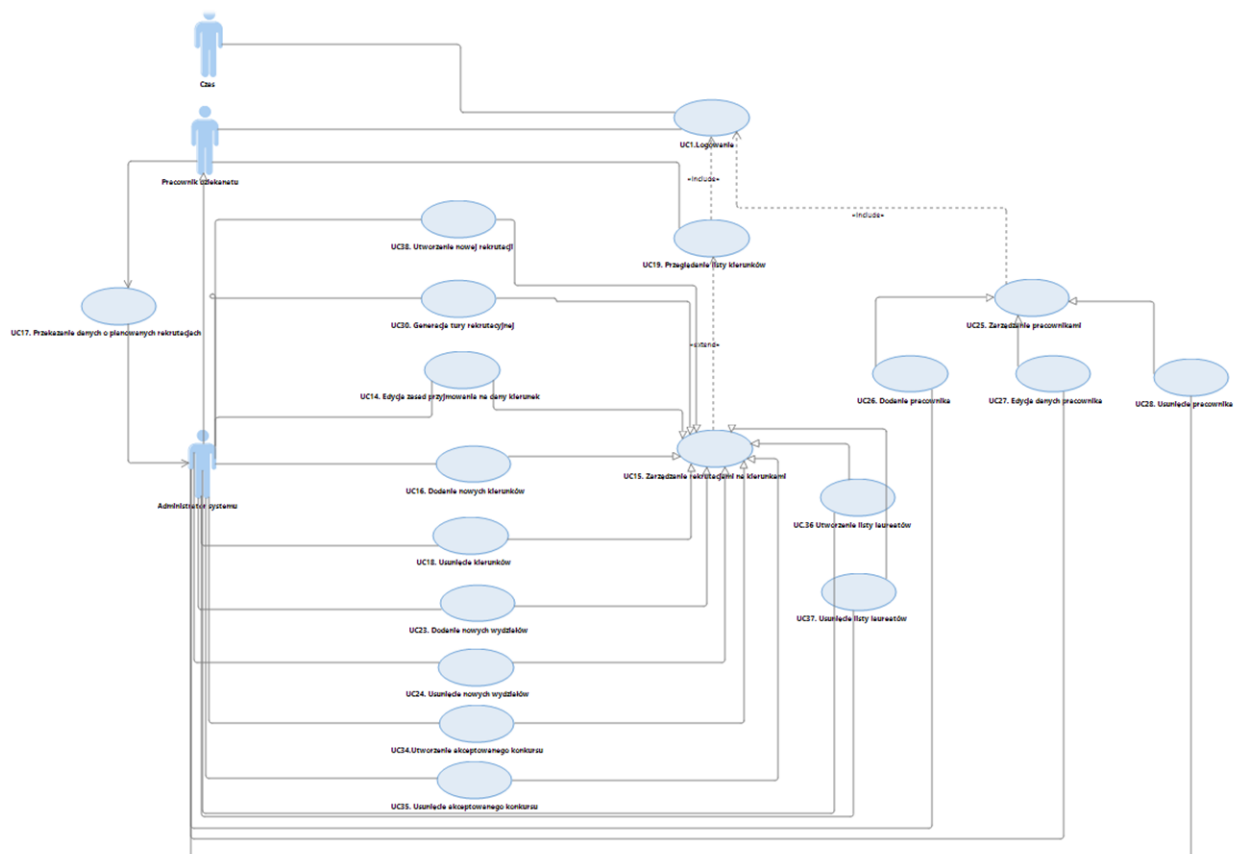


Diagram 4: Diagram przypadków użycia, pokazujący akcje możliwe do wykonania przez Pracownika albo Administratora Systemu podczas tworzenia oraz konfigurowania jeszcze nierozpoczętej tury rekrutacji.



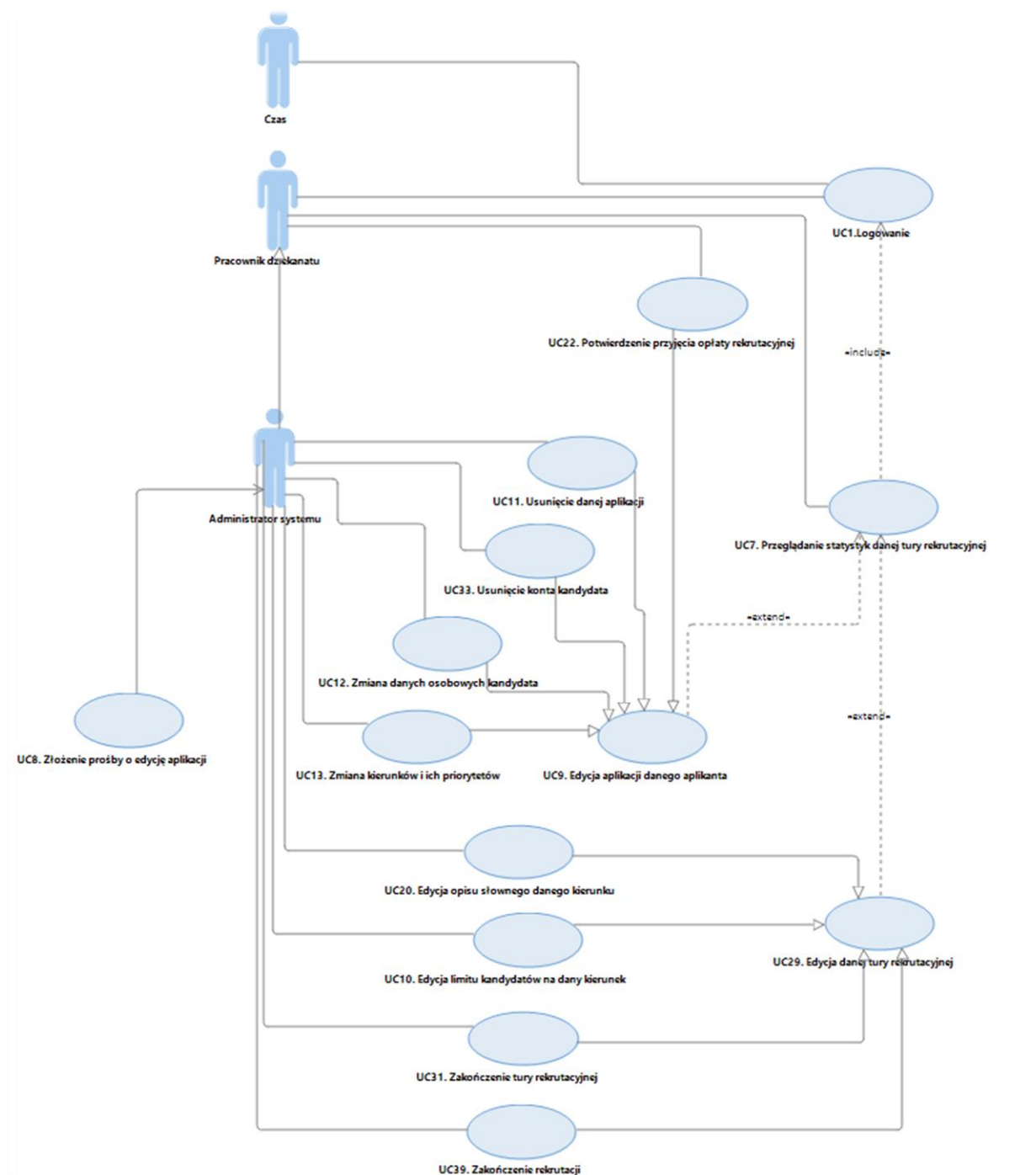


Diagram 5: Diagram przypadków użycia, przedstawiający akcje możliwe do wykonania przez Pracownika albo Administratora w trakcie trwania tury rekrutacji.

Opis przypadków użycia:

Przypadki użycia, podzielone na kategorie zależne od typu aktora oraz czasu wywołania:

**1. Kandydat, po zamknięciu rekrutacji:**

- UC1. Logowanie
- UC6. Odebranie informacji o zaakceptowanych aplikacjach
- UC7. Przeglądanie statystyk danej tury rekrutacyjnej

**2. Kandydat, w trakcie tury rekrutacji:**

- UC1. Logowanie
- UC2. Przeglądanie list dostępnych kierunków do rekrutacji
- UC3. Zarządzanie złożonymi aplikacjami
- UC4. Utworzenie nowej aplikacji na studia
- UC5. Opłacenie opłaty rekrutacyjnej za daną aplikację
- UC8. Złożenie prośby o edycję aplikacji
- UC32. Utworzenie konta kandydata na studia

**3. Pracownik, po zakończeniu rekrutacji:**

- UC1. Logowanie
- UC7. Przeglądanie statystyk danej tury rekrutacyjnej
- UC21. Pobranie list rekrutacyjnych

**4. Pracownik, przed rozpoczęciem rekrutacji:**

- UC1. Logowanie
- UC14. Edycja zasad przyjmowania na dany kierunek
- UC15. Zarządzanie rekrutacjami na kierunkami
- UC16. Dodanie nowego kierunku
- UC17. Przekazanie danych o planowanych rekrutacjach
- UC18. Usunięcie kierunku
- UC19. Przeglądanie listy kierunków
- UC23. Dodanie nowego wydziału
- UC24. Usunięcie wydziału
- UC25. Zarządzanie pracownikami
- UC26. Dodanie pracownika
- UC27. Edycja danych pracownika
- UC28. Usunięcie pracownika
- UC30. Generacja tury rekrutacyjnej
- UC34. Utworzenie akceptowanego konkursu
- UC35. Usunięcie akceptowanego konkursu
- UC36. Utworzenie listy laureatów
- UC37. Usunięcie listy laureatów
- UC38. Utworzenie nowej rekrutacji

**5. Pracownik, w trakcie tury rekrutacji:**

- UC1. Logowanie
- UC7. Przeglądanie statystyk danej tury rekrutacyjnej
- UC8. Złożenie prośby o edycję aplikacji
- UC9. Edycja aplikacji danego aplikanta
- UC10. Edycja limitu kandydatów na dany kierunek
- UC11. Usunięcie danej aplikacji
- UC12. Zmiana danych osobowych kandydata

- UC13. Zmiana kierunków i ich priorytetów
- UC20. Edycja opisu słownego danego kierunku
- UC22. Potwierdzenie przyjęcia opłaty rekrutacyjnej
- UC29. Edycja danej tury rekrutacyjnej
- UC31. Zakończenie tury rekrutacyjnej
- UC33. Usunięcie konta kandydata
- UC39. Zakończenie rekrutacji

## Scenariusze wykonania wybranych przypadków użycia

- **UC2. Przeglądanie list dostępnych kierunków do rekrutacji:**
- Aktorzy: Kandydat, Czas
- Warunki wejściowe:
  - Konieczne jest wykonanie przez kandydata przypadku UC1. Logowanie.
- Warunki wyjściowe:
  - Po wykonaniu przypadku użycia UC2 udostępniona zostaje kandydatowi do przejrzania lista kierunków.
- Scenariusz główny:
  1. Włączenie przypadku *UC1. Logowanie*, zależnego od *Czasu* i czy trwa obecnie tura rekrutacji na studia.
  2. Po pomyślnym uwierzytelnieniu i zakończeniu akcji UC1, wyświetlenie listy kierunków, na które *Kandydat* może wysłać aplikację. (patrz Storyboard rys.1 dla UC 2)
  3. Jeżeli zaznaczono opcję zarządzania wnioskami o przyjęcie na studia, wykonanie przypadku rozszerzającego *UC3. Zarządzanie złożonymi aplikacjami*. (patrz Storyboard rys.2 dla UC 2)
- **UC22. Potwierdzenie przyjęcia opłaty rekrutacyjnej:**
- Aktorzy: Pracownik dziekanatu
- Warunki wejściowe:
  - Konieczne jest wywołanie przez kandydata przypadku UC9. Edycja aplikacji danego aplikanta.
- Dane wejściowe:
  - Aplikacja danego Kandydata, którą dany Pracownik Dziekanatu ma potwierdzić poprzez zatwierdzenie przyjęcia płatności rekrutacyjnej.
- Scenariusz główny:
  1. Przypadek specjalizujący *UC9. Edycja aplikacji danego aplikanta*.
  2. Zaznaczenie opcji potwierdzenia przyjęcia opłaty rekrutacyjnej.
  3. Manualne sprawdzenie przez *Pracownika dziekanatu* wiarygodności przesłanego potwierdzenia opłacenia opłaty rekrutacyjnej.
  4. W wypadku zgodności danych, zaznaczenie danej aplikacji jako odpowiednio opłaconej.
- Scenariusz alternatywny:
  1. Taki sam jak w scenariuszu głównym
  2. Taki sam jak w scenariuszu głównym
  3. W przypadku braku zgodności danych, odrzucenie wysłanego potwierdzenia płatności i podtrzymanie statusu aplikacji jako aplikacji nieważnej.

- **UC3. Zarządzanie złożonymi aplikacjami:**
- Aktorzy: Kandydat
- Warunki wejściowe:
  - Konieczne jest wywołanie przez kandydata przypadku UC2. Przeglądanie list dostępnych kierunków do rekrutacji.
- Scenariusz główny:
  1. Rozszerzenie przypadku UC2. *Przeglądanie list dostępnych kierunków do rekrutacji po zaznaczeniu opcji zarządzania wnioskami o przyjęcie na studia.*
  2. Udostępnienie i wyświetlenie listy aktualnie prowadzonych aplikacji dla danego *Kandydata*.
  3. W zależności od wyboru *Kandydata* wywołanie jednej z opcji: UC4, UC5 lub UC8.
  
- **UC4. Utworzenie nowej aplikacji na studia:**
- Aktorzy: Kandydat
- Warunki wejściowe:
  - Konieczne jest wywołanie przez kandydata przypadku UC3. Zarządzanie złożonymi aplikacjami.
- Warunki wyjściowe:
  - Konieczne jest utworzenie klasy Aplikacja, przypisanej do danego Kandydata, przechowującej jego dane osobowe, preferencje co do wybieranych kierunków oraz potwierdzenie opłacenia opłaty rekrutacyjnej.
- Scenariusz główny:
  1. Przypadek specjalizujący UC3. *Zarządzanie złożonymi aplikacjami.*
  2. Zaznaczenie opcji utworzenia nowej aplikacji na studia.
  3. Wypełnienie przez *Kandydata* na studia formularza, pobierającego wszystkie potrzebne dane osobowe, takie jak numer PESEL, numer dokumentu tożsamości czy odległość od miejsca zamieszkania.
  4. Sprawdzenie poprawności wypełnienia formularza.
  5. W przypadku poprawnie wypełnionego formularza, wypełnienie przez *Kandydata* listy kierunków, na które chce złożyć prośbę o przyjęcie na studia i przypisanie im odpowiednich priorytetów.
  6. Przekazanie *Kandydatowi* informacji o poprawnie przyjętym formularzu i listy kierunków, oraz powiadomienie o konieczności uiszczenia opłaty rekrutacyjnej przez *Kandydata* w akcji UC5.
  7. Po otrzymaniu informacji od akcji UC5 o poprawnie uiszczonej opłacie rekrutacyjnej, dodanie aplikacji *Kandydata* do odpowiednich list rekrutacyjnych i obliczenie jego pozycji na tych listach, w zależności od priorytetów i liczby punktów rekrutacyjnych.
- Pierwszy scenariusz alternatywny:
  1. Taki sam jak w scenariuszu głównym.
  2. Taki sam jak w scenariuszu głównym.
  3. Taki sam jak w scenariuszu głównym.
  4. Po stwierdzeniu niepoprawnych danych w formularzu wypełnionym przez *Kandydata*, powrót do punktu 2. scenariusza głównego do czasu, gdy wszystkie pola formularza nie zostaną wypełnione poprawnie.
- Drugi scenariusz alternatywny:

1. Taki sam jak w scenariuszu głównym.
2. Taki sam jak w scenariuszu głównym.
3. Taki sam jak w scenariuszu głównym.
4. Taki sam jak w scenariuszu głównym.
5. Taki sam jak w scenariuszu głównym.
6. Stwierdzenie nieważności aplikacji złożonej przez *Kandydata* po *Czasie* zakończenia tury rekrutacji, jeżeli do tego czasu nie została potwierdzona opłata rekrutacyjna za pomocą akcji UC5.

- **UC5. Opłacenie opłaty rekrutacyjnej za daną aplikację:**

- Aktorzy: Kandydat
- Warunki wejściowe:
  - Konieczne jest wywołanie przez kandydata przypadku UC3. Zarządzanie złożonymi aplikacjami.
- Warunki wyjściowe:
  - Zapisanie potwierdzenia płatności w formie pliku w systemie, do zweryfikowania przez Pracownika dziekanatu.
- Scenariusz główny:
  1. Przypadek specjalizujący UC3. *Zarządzanie złożonymi aplikacjami*.
  2. Zaznaczenie opcji opłacenia opłaty rekrutacyjnej za daną aplikację.
  3. Przekazanie danych o przelewie, jaki *Kandydat* ma zrobić poza systemem, takie jak numer konta, nazwa odbiorcy i tytuł przelewu, a także kwota opłaty rekrutacyjnej.
  4. Udostępnienie możliwości wysłania przez *Kandydata* potwierdzenia płatności w formie pliku.
  5. Zapisanie w systemie potwierdzenia płatności w formie pliku po nadesłaniu go przez *Kandydata*.
  6. Jeżeli opłata rekrutacyjna została poprawnie wysłana (co sprawdza i uwierzytelnia *Pracownik* dziekanatu w akcji UC22. *Potwierdzenie przyjęcia opłaty rekrutacyjnej*), wysłanie odpowiedniej informacji do akcji UC4.

- **UC7. Przeglądanie statystyk danej tury rekrutacyjnej:**

- Aktorzy: Pracownik dziekanatu, Administrator, Czas
- Warunki wejściowe:
  - Konieczne jest wykonanie przez Pracownika dziekanatu lub Administratora przypadku UC1. Logowanie.
- Warunki wyjściowe:
  - Po wykonaniu przypadku użycia UC7 udostępniona zostaje pracownikowi dziekanatu albo administratorowi do przejrzania lista kierunków.
- Scenariusz główny:
  1. Włączenie przypadku UC1. *Logowanie*, zależnego od *Czasu* i czy trwa obecnie tura rekrutacji na studia.
  2. Po pomyślnym uwierzytelnieniu i zakończeniu akcji UC1, wyświetlenie statystyk obecnie prowadzonej tury rekrutacyjnej, takich jak np. liczba aplikacji na dany kierunek.  
(patrz Storyboard rys.1 dla UC 7)

3. Jeżeli zaznaczono opcję zarządzania aplikacjami, wykonanie przypadku rozszerzającego UC9. *Edycja aplikacji danego aplikanta.*
4. Jeżeli zaznaczono opcję edycji danej tury rekrutacyjnej lub kierunków do niej wprowadzonych, wykonanie przypadku rozszerzającego UC29. *Edycja danej tury rekrutacyjnej.*

- **UC8. Złożenie prośby o edycję aplikacji:**

- Aktorzy: Kandydat, Administrator systemu

- Opis:

- Złożenie prośby do administratora systemu, przez kandydata, o edycję jego aplikacji i takich jej części, jak np. wybrany kierunek studiów, dane osobowe studenta czy priorytety nadane poszczególnym kierunkom.

- Warunki wejściowe:

- Konieczne jest wykonanie przez kandydata przypadku UC4. Utworzenie nowej aplikacji na studia, by móc ją edytować

- Warunki wyjściowe:

- Po wykonaniu przypadku użycia UC7 przekazana zostaje administratorowi informacja o Kandydacie, chcącym złożyć prośbę o korektę jego aplikacji.

- Scenariusz główny:

1. Aktywacja przez *Kandydata* akcji UC8.
  - 1a. Uzupełnienie przez *Kandydata* wiadomości skierowanej do *Administratora systemu*, która specyfikuje, o jakiego rodzaju zmianę aplikacji *Kandydat* chce poprosić
  - 1b. *Kandydat* wybiera opcję wysłania zgłoszenie
  - 1c. Zgłoszenie zostaje wysłane do skrzynki odbiorczej *Administratora systemu*
2. Odebranie danych przez *Administratora systemu* z akcji UC8.
3. Wykonanie przez *Administratora systemu* stosownej akcji w zależności od treści zgłoszenia: UC11, UC12 lub UC13.

- Scenariusz alternatywny:

1. Taki sam jak w scenariuszu głównym
2. Taki sam jak w scenariuszu głównym
3. Odrzucenie zgłoszenia wysłanego przez *Kandydata*, gdy *Administrator systemu* stwierdzi, że dane zgłoszenie jest niezasadne lub niemożliwe do spełnienia.

- **UC9. Edycja aplikacji danego aplikanta:**

- Aktorzy: Pracownik dziekanatu, Administrator systemu

- Warunki wejściowe:

- Konieczne jest wywołanie przez kandydata przypadku UC7. Przeglądanie statystyk danej tury rekrutacyjnej.

- Scenariusz główny:

1. Rozszerzenie przypadku UC7. *Przeglądanie statystyk danej tury rekrutacyjnej* po zaznaczeniu opcji zarządzania aplikacjami.
2. Udostępnienie i wyświetlenie listy aktualnie prowadzonych aplikacji w systemie, w zależności od obecnie przeglądanej aplikacji w turze.
3. W zależności od wyboru *Administratora systemu* wywołanie jednej z opcji: UC22, UC11, UC12 lub UC13.
4. Wykonanie według wyboru *Pracownika dziekanatu* akcji UC22 jako jedynej dostępnej dla niego opcji edycji aplikacji.

## Diagramy Klas:

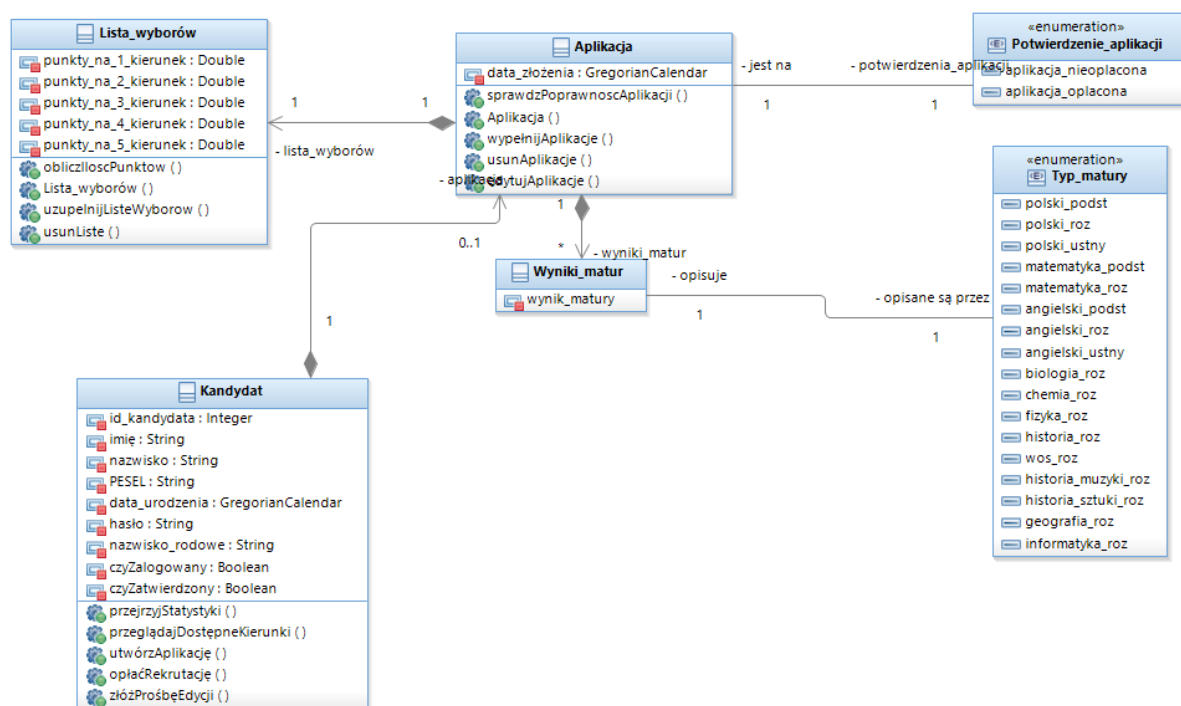


Diagram 6: Diagram klas, pokazujący obiekty przechowujące dane i operacje, które na tych danych można przeprowadzać we wprowadzanym systemie rekrutacyjnym. Diagram ten obejmuje klasy, służące do obsługi zapytań wydawanych przez aktora Kandydata.

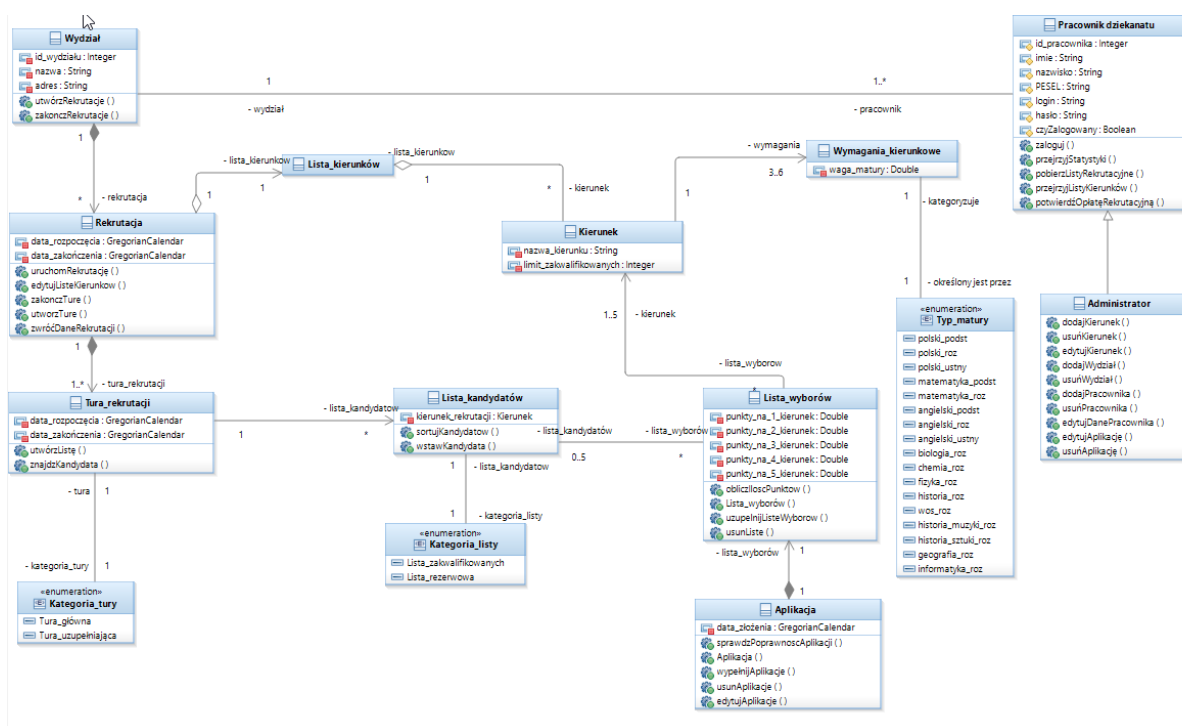


Diagram 7: Diagram klas, pokazujący obiekty przechowujące dane i operacje, które będzie można przeprowadzać na nich we wprowadzanym systemie rekrutacyjnym. Diagram ten obejmuje klasy, służące do obsługi zapytań wydawanych przez aktorów: Pracownika dziekanatu oraz Administratora.



## Diagramy Sekwencji:

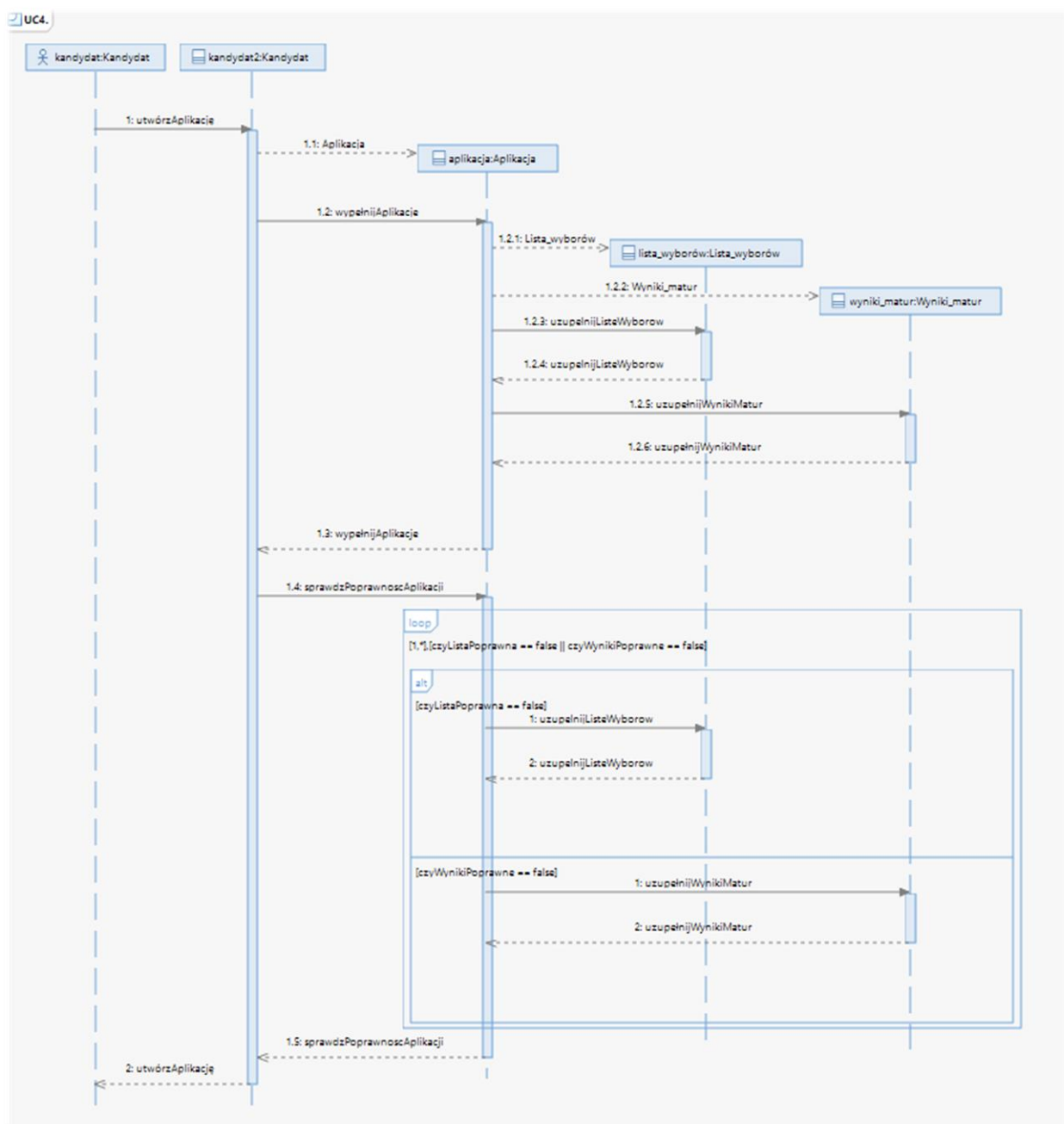


Diagram 8: Diagram sekwencji, przedstawiający podstawowy schemat komunikacyjny pomiędzy obiektami, pozwalający na realizację zapytania reprezentowanego przez przypadek użycia UC4.

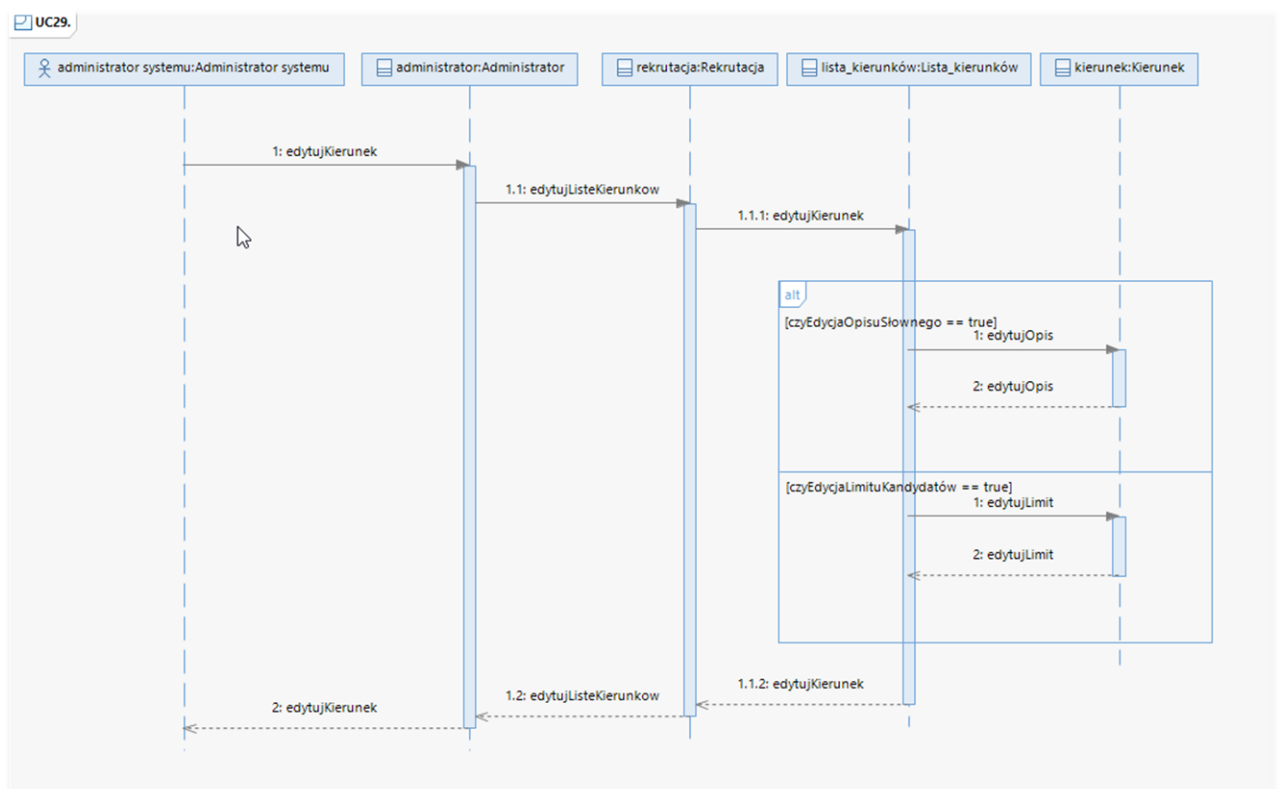


Diagram 9: Diagram sekwencji, przedstawiający podstawowy schemat komunikacyjny pomiędzy obiektami, pozwalający na realizację zapytania reprezentowanego przez przypadek użycia UC29.

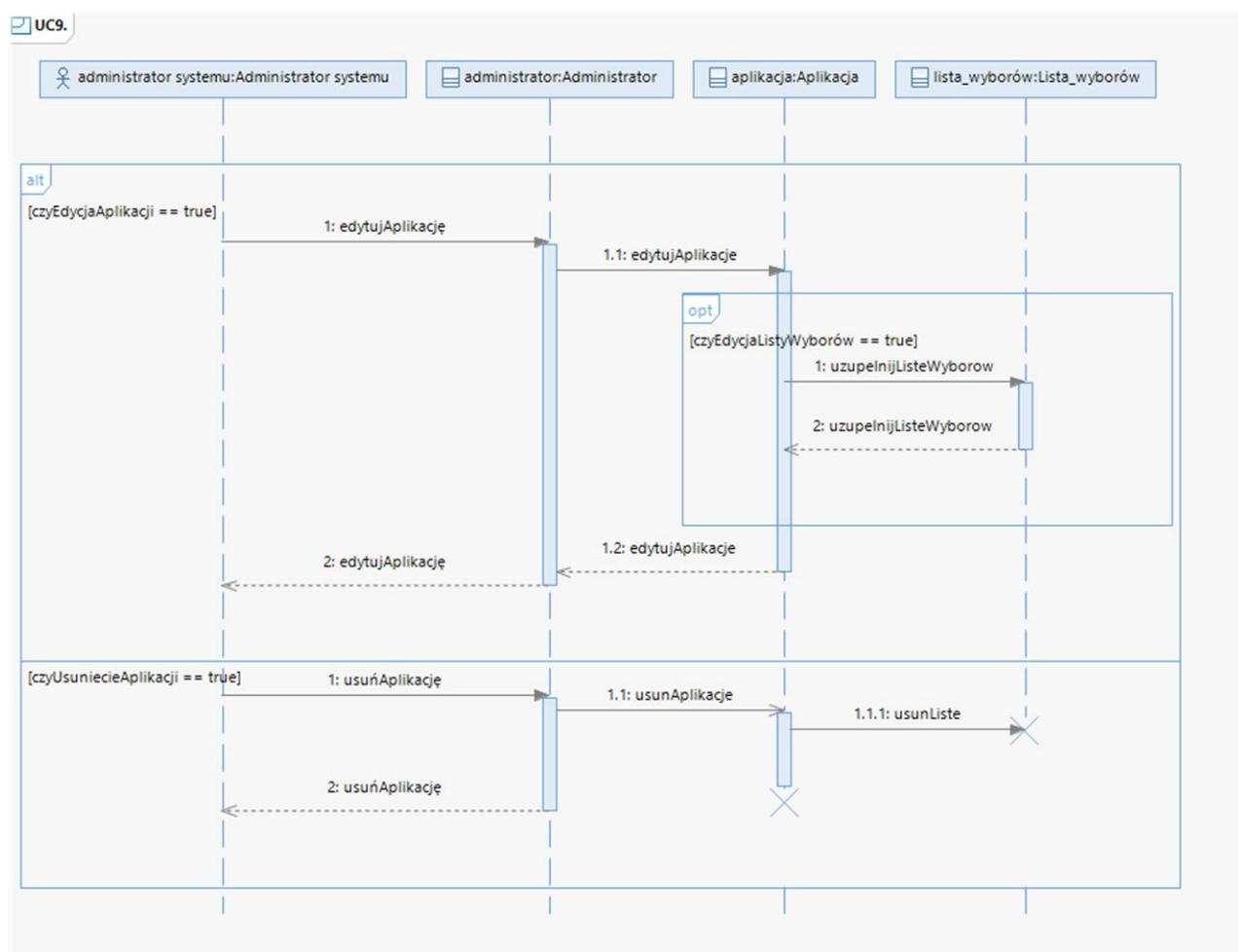


Diagram 10: Diagram sekwencji, przedstawiający podstawowy schemat interakcji pomiędzy obiektami w systemie, pozwalający na realizację zapytania reprezentowanego przez przypadek użycia UC9.

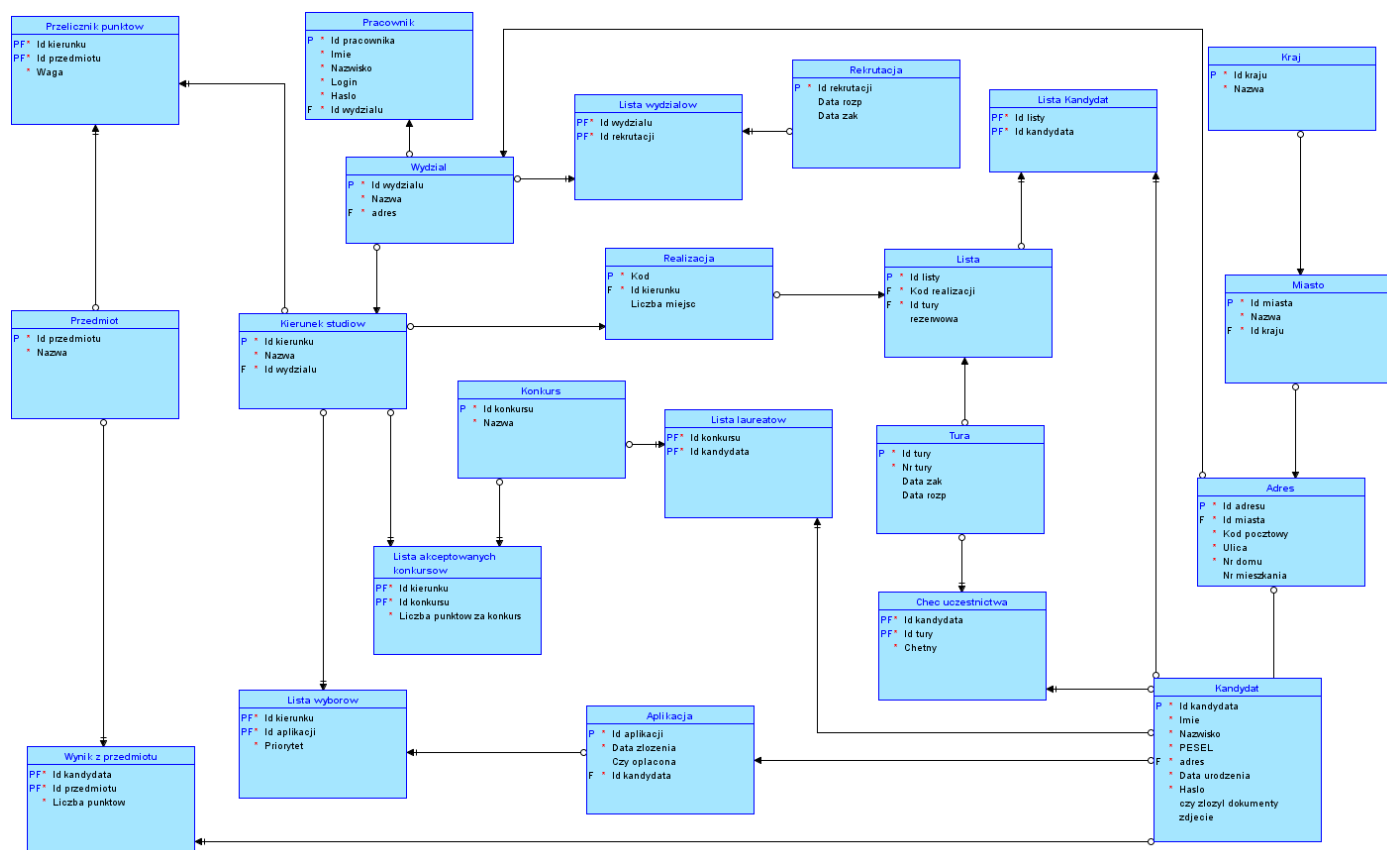
## Macierz CRUD - Łukasz Reszka:

Przypadek użycia \ Encja	APLIKACJA	CHEC_UCZESTNICTWA	KANDYDAT	KIERUNEK_STUDIOW	KONKURS	LISTA	LISTA_AKCEPTOWANYCH_KONKURSOROW	LISTA_LAUREATOW	LISTA_REZERWOWYCH	LISTA_WYBOROW	LISTA_WYDZIALOW	LISTA_ZAKWALIFIKOWANYCH	PRACOWNIK	PRZEDMIOT	PRZELICZNIK_PUNKTOW	REALIZACJA	REKRUTACJA	TURA	WYDZIAŁ	WYNIK_Z_PRZEDMIOTU
UC1. Logowanie			R										R					R		
UC2. Przeglądanie list dostępnych kierunków do rekrutacji				R							R						R		R	
UC3. Zarządzanie złożonymi aplikacjami	R			R						R	R					R	R	R	R	
UC4. Utworzenie nowej aplikacji na studia	CR	R	R	R	R	CR	R			CR	R			R	R	R	R	R	R	CR
UC5. Opłacenie opłaty rekrutacyjnej za daną aplikację	RU		R	R						R	R					R	R	R	R	
UC6. Odebranie informacji o zaakceptowanych aplikacjach	R		R	R		R			R	R		R				R		R	R	
UC7. Przeglądanie statystyk danej tury rekrutacyjnej	R	R	R	R	R	R	R	R	R	R	R	R				R	R	R	R	
UC8. Złożenie prośby o edycję aplikacji	R		R	R	R			R		R	R					R	R	R	R	
UC9. Edycja aplikacji danego aplikanta	R	R	R	R						R						R		R	R	
UC10. Edycja limitu kandydatów na dany kierunek				R							R					RU	R	RU	R	
UC11. Usunięcie danej aplikacji	RD	R	R	R						RD						R		R	R	
UC12. Zmiana danych osobowych kandydata	R	R	RU							R						R		R	R	
UC13. Zmiana kierunków i ich priorytetów	R	R	R	R						CRUD						R		R	R	
UC14. Edycja zasad przyjmowania na dany kierunek				R	R		CRUD				R			R	CRUD	R	R		R	
UC15. Zarządzanie rekrutacjami na kierunkach				R						R						R	R		R	
UC16. Dodanie nowego kierunku				CR	R		CR				R			R	CR	CR	R		R	
UC17. Przekazanie danych o planowanych rekrutacjach			R							R			R			R	R		R	
UC18. Usunięcie kierunku				RD		RD	RD			RD	R				RD	RD	R		R	
UC19. Przeglądanie listy kierunków				R												R			R	
UC20. Edycja opisu słownego danego kierunku				RU														RU	R	
UC21. Pobranie list rekrutacyjnych						R			R			R						R		
UC22. Potwierdzenie przyjęcia opłaty rekrutacyjnej	RU	R	R	R						R						R		R	R	
UC23. Dodanie nowego wydziału				R							CR					R	R		CR	
UC24. Usunięcie wydziału				RU							RD		RU			R	R		RD	
UC25. Zarządzanie pracownikami													R						R	
UC26. Dodanie pracownika													CR						R	
UC27. Edycja danych pracownika													RU						R	
UC28. Usunięcie pracownika													RD						R	
UC29. Edycja danej tury rekrutacyjnej																		RU		

Tabela 1: Tabela przedstawiająca macierz CRUD, pozwalającą na określenie zależności pomiędzy obiektami w bazie danych a przypadkami użycia, które je modyfikują.

## Model E-R - Jakub Kordel, Jakub Strawa:

Notacja Bachmana:



Notacja barkera:

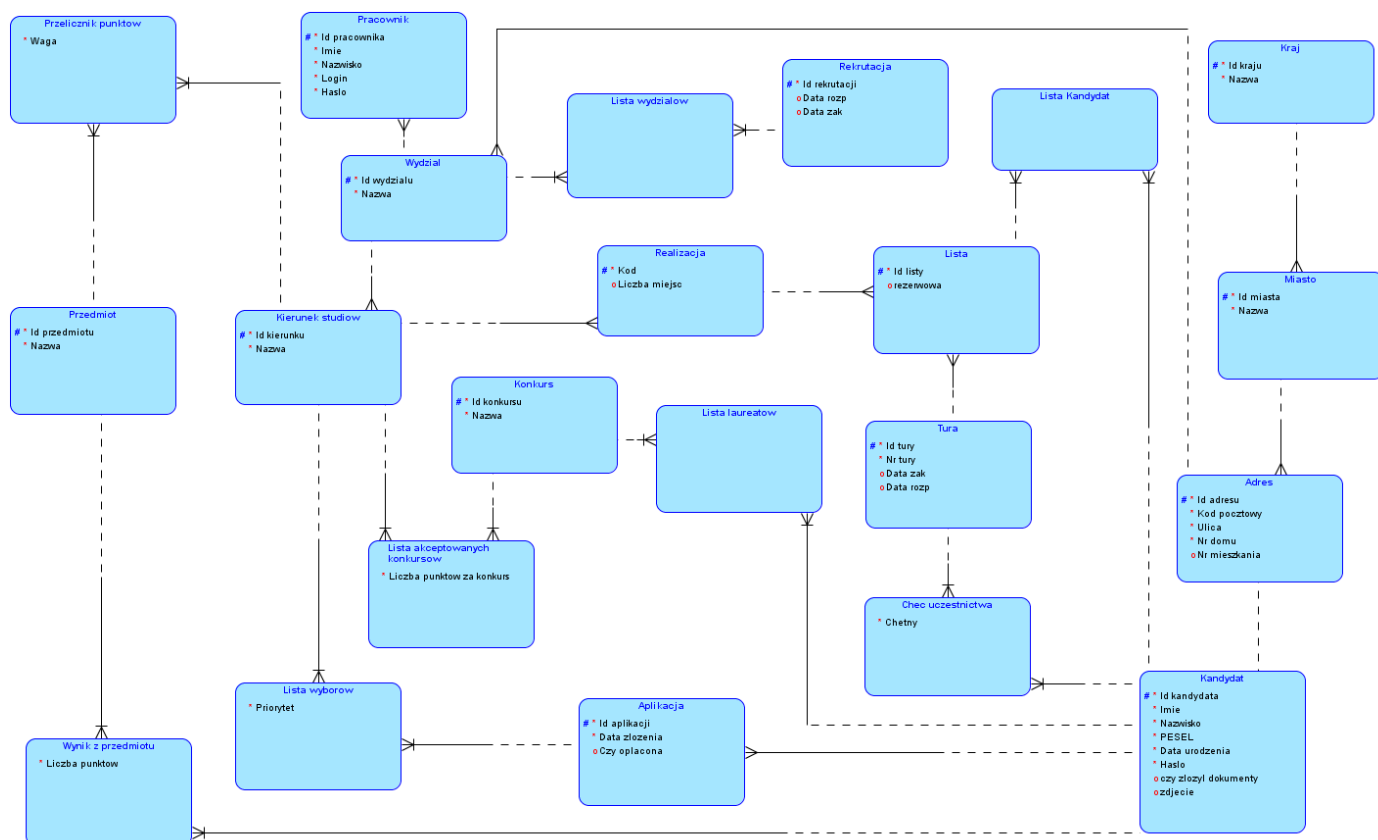


Diagram 11: Diagram przedstawiający model E-R powstającej bazy danych, przechowującej informacje o Kandydatach, ich preferowanych kierunkach, jak również i turach rekrutacji, przeprowadzanych przez dane jednostki administracyjne dla Kandydatów.

## Model Logiczny - Jakub Kordel:

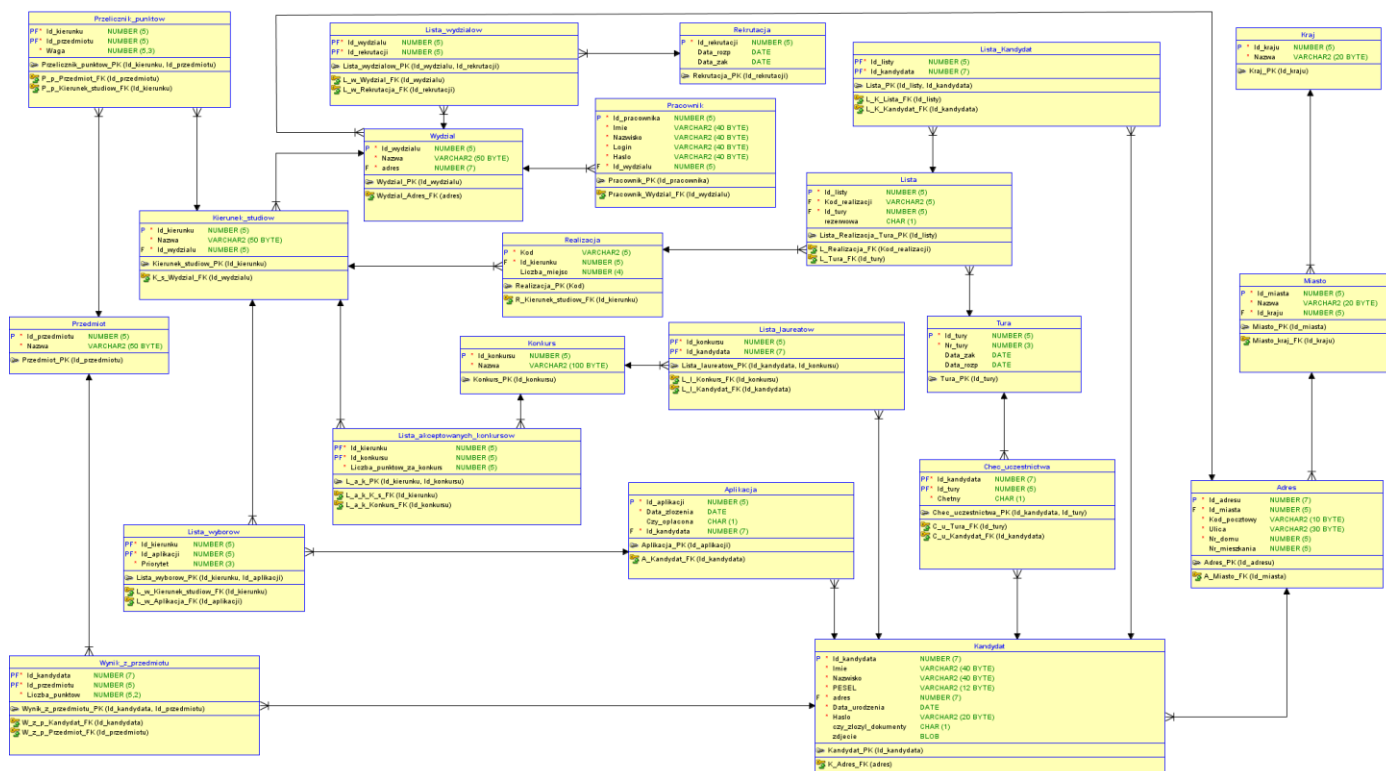


Diagram 12: Diagram przedstawiający model relacyjny powstającej bazy danych.

Przykład funkcji obliczającej liczbę punktów kandydata na dany kierunek (w języku PL/SQL):

```

CREATE OR REPLACE FUNCTION oblicz_punkty_kandydata(kan_id NUMBER, kierunek_id NUMBER)
RETURN NUMBER
AS
    punkty NUMBER := 0;
    z_przedm NUMBER := 0;
    z_konk NUMBER := 0;
    przel_punktow przelicznik_punktow%rowtype;
    konkurs_kandydata lista_laureatow%rowtype;
    CURSOR liczone_przedmioty(kier_id NUMBER)
    IS
        SELECT *
        FROM przelicznik_punktow
        WHERE id_kierunku = kier_id;

    CURSOR konkursy_kandydata(kand_id NUMBER)
    IS
        SELECT *
        FROM lista_laureatow
        WHERE id_kandydata = kand_id;

BEGIN
    OPEN liczone_przedmioty(kierunek_id);
    LOOP
        FETCH liczone_przedmioty INTO przel_punktow;
        EXIT WHEN liczone_przedmioty%notfound;
        select liczba_punktow into z_przedm from wynik_z_przedmiotu where id_przedmiotu = przel_punktow.id_przedmiotu and id_kandydata = kan_id;
        IF z_przedm != null THEN
            punkty := punkty + z_przedm*przel_punktow.waga;
        END IF;
    END LOOP;
    CLOSE liczone_przedmioty;

    OPEN konkursy_kandydata(kan_id);
    LOOP
        FETCH konkursy_kandydata INTO konkurs_kandydata;
        EXIT WHEN konkursy_kandydata%notfound;
        select liczba_punktow_za_konkurs into z_konk from lista_akceptowanych_konkursow where konkurs_kandydata.id_konkursu = id_konkursu and id_kierunku = kierunek_id;
        IF z_konk != null THEN
            punkty := punkty + z_konk;
        END IF;
    END LOOP;
    CLOSE konkursy_kandydata;

    RETURN punkty;
END;

```

Aktywui s

## Przykład procedury tworzącej aplikacje kandydata (w języku PL/SQL):

```

CREATE OR REPLACE PROCEDURE zloz_aplikacje(kand_id NUMBER, kier1 NUMBER, kier2 NUMBER, kier3 NUMBER, kier4 NUMBER, kier5 NUMBER)
AS
    apl_id NUMBER;
BEGIN
    select max(id_aplikacji)+1 into apl_id from aplikacja;
    INSERT INTO aplikacja (id_aplikacji, data_zlozenia, czy_oplacona, id_kandydata)
    VALUES (apl_id, CURRENT_DATE, 0, kand_id );

    INSERT INTO lista_wyborow (id_kierunku, id_aplikacji, priorytet)
    VALUES (kier1, apl_id, 1);
    INSERT INTO lista_wyborow (id_kierunku, id_aplikacji, priorytet)
    VALUES (kier2, apl_id, 2);
    INSERT INTO lista_wyborow (id_kierunku, id_aplikacji, priorytet)
    VALUES (kier3, apl_id, 3);
    INSERT INTO lista_wyborow (id_kierunku, id_aplikacji, priorytet)
    VALUES (kier4, apl_id, 4);
    INSERT INTO lista_wyborow (id_kierunku, id_aplikacji, priorytet)
    VALUES (kier5, apl_id, 5);
END;

```



## Model fizyczny – Jakub Kordel:

Poniżej znajduje się analiza modelu logicznego w kontekście jego charakterystyk fizycznych. Pod analizą zamieściliśmy model – fizyczny, który jest wersją po denormalizacji modelu logicznego.

### Szacowane wielkości tabel:

- Wydział: <40 rekordów
- Kierunek\_studiów: <100 rekordów
- Rekrutacja:

Uczelnia prowadzi 2 rekrutacje rocznie, więc po 5 latach ~ 10 rekordów.

- Pracownik:

Uczelnia zatrudnia około 5tyś. Pracowników, trzeba założyć, że <7000 rekordów.

- Kandydat:

W trakcie dwóch rekrutacji na semestr zimowy i letni liczba aplikujących wynosi nie więcej niż 100tyś. Jeżeli po zakończonej rekrutacji dane kandydatów nie będą usuwane to trzeba liczyć, że po 5 latach liczba rekordów może urosnąć do 0.5mln.

- Lista:

By oszacować z góry założę, że nabór podczas danej rekrutacji jest prowadzony przez wszystkie kierunki oferowane przez uczelnię. Każdy kierunek może przeprowadzić 3 tury podczas rekrutacji, po pierwszych z dwóch tur generowana jest lista zakwalifikowanych jak i rezerwowych a w ostatniej turze tylko lista zakwalifikowanych. Zatem liczba rekordów =  $5 * \text{liczba kierunków} < 500$  list podczas 1 rekrutacji. Po 5 latach ~ 5000 rekordów (zakładając, że rekrutacja jest prowadzona 2 razy w roku).

- Lista\_Kandydat:

Tabela łącząca listę z kandydatem. Zakładając, że liczba miejsc na kierunki wynosi średnio 100 a liczba miejsc na pierwszej liście rezerwowych 40, drugiej 20, to trzeba liczyć około 200 rekordów na kierunek podczas jednej 3-turowej rekrutacji. Podczas jednej rekrutacji trzeba liczyć ~20tyś. rekordów. Po 5 latach ~200tyś.

- Konkurs: ~1000
- Lista\_laureatów:

Zakładając, optymistycznie, że na 10 kandydatów przypada jeden laureat konkursu: ~10tyś. rekordów na rok. Aczkolwiek będzie to prawdopodobnie znacznie mniej.

- Lista\_akceptowanych\_konkursów:

Licząc, że kierunek nie liczy punktów za więcej niż 20 różnych konkursów:  $\sim 20 * 100 = 2$ tyś. rekordów

- Tura:

Liczba tur na rekrutacje: 3. Dwie rekrutacje rocznie dają około 30 rekordów po 5 latach.

- Realizacja:

Nie więcej niż 1 realizacja kierunku w danej rekrutacji. Podczas jednej rekrutacji zatem nie przybędzie więcej niż 100 rekordów. Po 5 latach < 1tyś. rekordów.

- Aplikacja:

Porównywalna z liczbą kandydatów, < 100tyś. rocznie.

- Cheć\_uczestnictwa:

Kandydat może zgłosić chęć uczestnictwa w drugiej i w trzeciej turze rekrutacji. <200tyś. rekordów rocznie.

- Przedmiot: <50 rekordów
- Wynik\_z\_przedmiotu:

Zakładając, że kandydat wprowadzi średnio niż 10 wyników maturalnych ~1mln rekordów rocznie.

- Przelicznik\_punktów:

Zakładając, że na średnio na jednym kierunku liczą się punkty z 10 przedmiotów to: < 1tyś. rekordów.

- Kraj: < 300 rekordów
- Miasto:

Jest 1tyś. miast w Polsce, kandydatów zagranicznych będzie mniej, ale za to miasta będą się mniej powtarzały. Trzeba liczyć, że może urosnąć do 10tyś. rekordów.

- Adres:

Tyle ile kandydatów, ~100tyś. nowych adresów rocznie.

- Lista\_wydziałów:

Jest to lista wydziałów biorących udział w danej rekrutacji. Zakładając, że każdy wydział bierze udział w rekrutacji, jest to około 30 rekordów w trakcie jednej rekrutacji.

- Lista\_wyborów:

Każda aplikacja może zawierać do 5 wyborów kierunku. Zatem szacując z góry: liczba aplikacji \* 5. około 500tyś. rekordów rocznie.

### Analiza częstotliwości wykonywania operacji:

Poniżej przedstawiona jest lista operacji, które będą często wywoływane:

- Wyszukiwanie kierunków dostępnych w danej rekrutacji (select)

Jednocześnie w systemie może być zalogowanych nawet 2500 kandydatów. Duża część z nich prawdopodobnie będzie przeglądać listę kierunków. Częstość wykonywania tej operacji może zatem osiągnąć nawet 1000 zapytań na minutę. Liczba kierunków na szczęście nie jest duża. Kandydaci będą szukać szczegółowych informacji o wydziale czy kierunku po ich nazwach.

- Rejestracja kandydata (insert)

Podczas jednej rekrutacji liczba kandydatów wynosi około 60tyś. Większość z nich będzie rejestrowała się podczas pierwszego tygodnia rekrutacji. Trzeba liczyć, że częstość wykonywania tej operacji osiągnie 500/na godzinę.

- Wyszukiwanie kandydatów po imieniu i nazwisku (select).

Pracownik dziekanatu często będzie potrzebował wyszukiwać kandydatów po imieniu i nazwisku np. po otrzymaniu potwierdzenia wpłaty kandydata, który nie podał swojego identyfikatora w tytule przelewu lub gdy kandydat zapomniał swojego identyfikatora i chce go odzyskać. Biorąc pod uwagę liczbę kandydatów słuszne jest założenie indeksu na kolumny imienia i nazwiska kandydata.

- Edycja, zarządzanie aplikacją przez kandydata (insert, delete)

Kandydaci w trakcie rekrutacji mają możliwość edytowania swojej aplikacji (wybranych kierunków i ich priorytetów). Z tego względu oraz biorąc pod uwagę liczbę aplikujących kandydatów można spodziewać się częstych operacji insert i delete na tabelach aplikacja i lista\_wyborów.

## Model

Poniższy diagram powstał po denormalizacji modelu logicznego. Na podstawie analiz dotyczących częstości operacji oraz w celu uproszczenia architektury aplikacji i przyspieszenia zapytań do bazy zdecydowaliśmy się na zmiany:

- Usunięcie tabeli Lista\_wyborów

W celu przyspieszenia zapytań i modyfikacji preferencji kandydatów zdecydowaliśmy się na usunięcie tabeli Lista\_wyborów, której celem było trzymanie informacji o preferencjach w osobnej tabeli niż tabela Aplikacja. Osobna tabela miała na celu oszczędzenie pamięci w przypadku, gdy kandydat nie skorzysta ze wszystkich przysługujących mu wyborów oraz zapewnienia braku potrzeby edycji kolumn tabeli Aplikacja w przypadku, gdy uczelnia zmieni ilość przysługujących wyborów. Uczelnia jednak, będzie oferowała 5 okien wyboru preferencji kierunków, a w razie zmiany tej decyzji koszt modyfikacji tabeli nie będzie duży, dlatego zdecydowaliśmy się na trzymanie informacji o wyborach w kolumnach tabeli Aplikacja.

- Usunięcie tabeli Lista\_wydziałów

Tabela miała na celu przechowywanie informacji o wydziałach biorących udział w danej rekrutacji. Korzystanie z tej informacji w celu znalezienia realizacji kierunków biorących udział w danej rekrutacji wymagało by jednak wielu skomplikowanych zapytań do bazy. Dlatego zamiast takiego rozwiązania zdecydowaliśmy się na trzymanie informacji o rekrutacji w tabeli Tura a turę łączyć z każdą realizacją w tabeli lista.

- Dodanie tabeli zgłoszenie

Dodaliśmy również nową tabelę zgłoszenie, która ma na celu przechowywanie zgłoszeń błędów dodanych przez kandydatów. Informacja o błędzie przechowywana jest w kolumnie opis.




## Storyboard'y – Jakub Strawa:

Przeglądanie dostępnych kierunków UC2.

rys.1

Uczelnia

Obecnie trwa rekrutacja na semestr: **wiosna 2021, tura 1**  
Do końca obecnej tury pozostało: **50 dni, 8 godzin, 20 minut**









Zalogowano jako: **Jan Kowalski** 

Przeglądaj dostępne kierunki

Zarządzaj preferencjami

Wydział: 

Wydział Informatyki

Kierunek	Liczba miejsc	Specjalizacje	Ustaw preferencje
Informatyka	100	SID, ISI	0  
Automatyka i Robotyka	50	AIR	0  
Telekomunikacja	200	SPEC1, SPEC2	0  
Elektronika	170	SPEC1, SPEC2, SPEC3	0  

Strona wydziału: [www.wydzialinformatyki.pl](http://www.wydzialinformatyki.pl)


rys.2

Uczelnia

Obecnie trwa rekrutacja na semestr: **wiosna 2021, tura 1**

Do końca obecnej tury pozostało: **50 dni, 8 godzin, 20 minut**

Zalogowano jako: **Jan Kowalski**



Zarządzaj preferencjami wyboru kierunków

Kierunek	Wydział	Numer preferencji
Administracja	Wydział Zarządzania	5
Informatyka	Wydział Informatyki	2
Automatyka i Robotyka	Wydział Informatyki	1
Mechatronika	Wydział Mechatroniki	3
Fizyka Teoretyczna	Wydział Fizyki	4

Przeglądanie statystyk i edytowanie informacji o rekrutacji UC7.


Rys.1

Uczelnia

Obecnie trwa rekrutacja na semestr: **wiosna 2021, tura 1**

Do końca obecnej tury pozostało: **50 dni, 8 godzin, 20 minut**

Zalogowano jako: **Admin**



Statystyki tury

Liczba wszystkich aplikacji	1234
Liczba wniosków do aplikacji	47
Liczba aplikacji biorących udział w turze	740

Liczba kandydatów na miejsce

Wydział Fizyki

Fizyka Teoretyczna

Liczba kandydatów: 60, liczba miejsc: 40, liczba kandydatów na 1 miejsce: 1,33.

Obecna tura rekrutacji

Wydział Fizyki


Fizyka Teoretyczna

Numer tury

1 z 5


Data zakończenia tury

20.08.2021



Data zakończenia rekrutacji

10.09.2021



Edytuj

September 2021

Mo	Tu	We	Th	Fr	Sa	Su
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					

## Wymagania нефункционалне – Jakub Kordel:

- W trakcie rekrutacji, liczba kandydatów może osiągnąć około 60 tysięcy. Zakładając, że kandydat spędza codziennie godzinę pracując z systemem, wymagane jest, żeby system był w stanie efektywnie obsługiwać (czas odpowiedzi systemu nie większy niż 4 sekundy) jednocześnie co najmniej 2500 kandydatów.
- Użytkowników będących pracownikami dziekanatu będzie znacznie mniej w porównaniu do kandydatów. System powinien obsługiwać pracowników dziekanatu z priorytetem. Bez względu na obciążenie systemu przez kandydatów, dostęp dla pracowników dziekanatu powinien pozostać szybki.
- Interfejs graficzny aplikacji powinien być łatwy w obsłudze dla kandydata jak i pracownika dziekanatu.
- Dane kandydatów takie jak numer PESEL muszą być wyjątkowo chronione, system musi przestrzegać przepisów RODO.
- Aplikacja pozwalająca na dostęp do systemu powinna posiadać wersje uruchamialną na każdym z wymienionych systemów operacyjnych: Windows, Linux, MacOS, Android, iOS.
- Wymagana jest wysoka niezawodność systemu. W razie awarii lub wystąpienia błędu, wymagana jest naprawa w ciągu 2 dni. Awaria systemu może spowodować potrzebę zmiany daty końca np. tury rekrutacji co jest bardzo niepożądane.
- System powinien być przenoszalny, oczekiwana jest ewentualna możliwość eksportu danych.