Akademia Górniczo-Hutnicza
im. Stanisława Staszica w Krakowie

Wydział Informatyki, Elektroniki i Telekomunikacji

KATEDRA INFORMATYKI

DOKUMENTACJA TECHNICZNA

TOMASZ KASPRZYK, DANIEL OGIELA, JAKUB STĘPAK

SYSTEM OBLICZAJĄCY WYNIKI WYBORÓW DLA UOGÓLNIENIA SYSTEMU K-BORDA

PROMOTOR:
dr hab. inż. Piotr Faliszewski

Kraków 2016

Spis treści

1. Dziedzina problemu

1.1. Metoda obliczania wyników wyborów

1.1.1. Metoda Bordy

Niech v będzie głosem nad zbiorem kandydatów C. Wynik według Bordy kandydata c w v jest równy $\beta(i) = C - i$, gdzie i- pozycja kandydata w ciągu v. Wynik c w wyborach jest sumą wyników c u każdego z wyborców

1.1.2. Metoda k-Borda

Rozszerzenie metody Bordy. Wynik, zamiast dla jednego kandydata, obliczany jest dla ciągu kandydatów. f_{kB}- funkcja zadowolenia z komitetu. Ciąg $(i_1, \ldots, i_k)$- ciąg pozycji kandydatów

Przyklad $C = c_1, c_2, c_3, c_4$ - zbiór kandydatów, $v = (c_2, c_1, c_4, c_3)$ - głos Niech $k = 2$ (wybory 2 spośród 4) $w = (c_4, c_3)$ Najpierw określamy pozycje kandydatów z komitetu w w v: $pos_v(w) = (3, 4)$, zatem wynik komitetu w dla głosu v wynosi $f_{kB}(3, 4) = (3)+(4) = ||C||-3)+(||C||-4) = 1+0 = 1$

1.1.3. Uogólnienie - system ℓ_p Borda

Zanim wprowadzone zostanie pojęcie uogólnionego systemu k-Borda warto przypomnieć wzór na normę ℓ_p

Norma ℓ_p

$\ell_p(x_1, x_2, \ldots, x_n) = \sqrt[p]{x_1^p + x_2^p + \ldots + x_n^p}$,
Wówczas, w uogólnionej wersji metody k-Borda, funkcja zadowolenia f_{kB} zostaje uzależniona również od parametru p z powyższego wzoru. Norma liczona jest z wyników według Bordy, $\beta(i)$. Wzór uogólniony funkcji zadowolenia przyjmuje zatem postać:
$f_{\ell_p B}(p, (i_1 1, \ldots, i_k)) = p[(i1)]p + [(i2)]p + ... + [(ik)]p$
Systemy k-Borda i Cahmberlin'a-Courant'a są szczególnymi przypadkami zdefiniowanego powyżej systemu ℓ_p- Borda:
Dla $p = 1, l_1$
$f_{\ell_p B}(1, (i_1, \ldots, i_k)) = \beta(i_1) + \beta(i_2) + \ldots + \beta(ik) = f_{kB}(i_1 1, \ldots, i_k)$
Dla $p = \infty, l_\infty = max$
$f_{\ell_p B}(\infty, (i_1, \ldots, i_k)) = \lim_{p\to\infty} \sqrt[p]{\beta\left[(i_1)\right]^p + \beta\left[(i_2)\right]^p + \ldots + \left[\beta(i_k)\right]^p} = \max \beta(i_1), \beta(i_2), \ldots, \beta(i_k) = \beta(i_1) = f_{CC}$

1.2. Format danych wejściowych

Pojedynczy plik składa się z następującego formatu:

```
<liczba kandydatów>
1, <nazwa kandydata>
2, <nazwa kandydata>
...
<liczba kandydatów>, <nazwa kandydata>
<liczba głosujących>, <liczba głosów policzonych>, <liczba unikalnych
głosów>
<liczba powtórzeń głosu>, <głos>
...
<liczba powtórzeń głosu>, <głos>
```

2. Architektura Django

$Django$ to framework webowy napisany w $Pythonie$. Dostarcza wysokopoziomowych abstrakcji pozwalających na szybkie i wygodne pisanie przejrzystych aplikacji.

Jego architektura koncepcyjnie przypomina wzorzec architektoniczny $Model - View - Controller$, jednak jak przyznają sami twórcy [ref], $Django$ nie do końca wpasowuje się w klasyczne ujęcie MVC.

$Django$ dostarcza mapowania obiekto-relacyjnego, dzięki którym całość modeli można ująć w $Pythonie$. Wygodny ORM zazwyczaj wystarcza do obsługi bazy danych, jednak zawsze istnieje możliwość użycia bezpośrednio SQL.

Widoki w $Django$ spełniają dwojaką funkcję - służą zarówno przekazaniu danych do wyświetlenia, jak i ich modyfikacji. W wyświetleniu danych użytkownikowi pośredniczą szablony ($templates$), które ópakowują"przekazane dane do postaci $HTMLa$, który może wyświetlić przeglądarka internetowa. Dzięki temu wybór danych, jakie mają zostać pokazane użytkownikowi jest oddzielony od samego sposobu ich prezentacji.

Za kontroler z klasycznego MVC można uznać sam framework dostarczający wspomnianej obsługi bazy danych czy mapowania adresów URL do poszczególnych widoków.

3. Opis modułów

3.1. Moduł administracji kont

3.1.1. Opis ogólny

Moduł odpowiedzialny za zarządzanie kontami użytkowników. Umożliwia czynności logowania, wylogowania oraz rejestracji. Nie współpracuje z innymi modułami. Korzysta jedynie z bazy danych w celu weryfikacji użytkowników.

3.1.2. Komponenty programowe

Komponenty programowe dla tego modułu znajdują się w pakiecie $ecs.accounts$

Typ komponentu	Komponenty	Wykorzystanie
widoki	klasa $LoginView$	logowanie
	klasa $RegisterView$	rejestracja
szablony	$login.html$	logowanie
	$register.html$	rejestracja
formularze	klasa $LoginForm$	logowanie
	klasa $RegistrationForm$	rejestracja

3.2. Moduł zarządzania wyborami

3.2.1. Opis ogólny

Moduł odpowiedzialny za administrację wyborami i ich wynikami. Umożliwia podstawowe operacje na wyborach i ich wynikach oraz nawigację między nimi. Pozwala na wyświetlenie listy wszystkich wyborów i ich wyników, stworzenie nowych wyborów lub wyniku, czy usunięcie wyborów. W celu wykonania niektórych zadań współpracuje z modułem obliczania wyników wyborów oraz modułem wizualizacji wyborów i ich wyników. Jest to najbardziej rozbudowany moduł.

3.2.2. Komponenty programowe

Typ komponentu	Komponenty	Wykorzystanie
widoki	klasa $ElectionListView$	wyświetlenie listy wszystkich wyborów
	klasa $ElectionCreateView$	stworzenie wyborów
	klasa $ElectionDeleteView$	usunięcie wyborów
	klasa $ElectionDetailView$	wyświetlenie informacji szczegółowych o danych wyborach
	klasa $ResultCreateView$	stworzenie wyniku dla danych wyborów - wyświetlenie formularza określającego parametry wyniku
	klasa $ResultDetailsView$	wyświetlenie danego wyniku danych wyborów
	klasa $ResultDeleteView$	usuwanie pojedynczego wyniku wyborów
szablony	$election_list.html$	wyświetlenie listy wszystkich wyborów, usunięcie wyborów - strona sukcesu wyświetlana po usunięciu wyborów
	$election_create.html$	stworzenie wyborów
	$election_details.html$	stworzenie wyborów - strona sukcesu wyświetlana po stworzeniu wyborów, wyświetlenie informacji szczegółowych o danych wyborach
	$election_delete.html$	usunięcie wyborów
	$result_create.html$	stworzenie nowego wyniku wyborów - strona z formularzem
	$result_details.html$	wyświetlenie danego wyniku danych wyborów
	$result_delete.html$	potwierdzenie usunięcia rezultatu
formularze	klasa $ElectionForm$	stworzenie wyborów
	klasa $ResultForm$	stworzenie wyniku dla danych wyborów - formularz określający parametry wyniku

3.3. Moduł wizualizacji wyborów i wyników wyborów

3.3.1. Opis ogólny

Moduł odpowiedzialny za stworzenie wykresu $2D$ wizualizującego wybory i jego wyniki. Wizualizacja dotyczy tylko wyborów wygenerowanych z rozkładu normalnego. Wyborcy i kandydaci są reprezentowani jako punkty na płaszczyźnie. Punkty reprezentujące wyborców i kandydatów mają na wykresie odmienne kolory. Na wykresie wyników wyborów punkty reprezentujące zwycięzców wyborów są powiększone. Moduł współpracuje z modułem zarządzania wyborów, który zleca mu zadanie wizualizacji wyborów lub jego wyników. W celu wykonania zadania moduł wizualizacji wyborów i wyników wyborów pobiera dane z bazy danych.

3.3.2. Komponenty programowe

Typ komponentu	Komponenty	Wykorzystanie
widoki	klasa $ScatterChartMixin$	pobranie danych potrzebnych do wygenerowania wykresów (punkty, tytuł wykresu, serii danych), dziedziczy po klasie $View$
	klasa $ElectionChartView$	wizualizacja wyborów, klasa dziedziczy po klasie $ScatterChartMixin$, pobranie współrzędnych kandydatów i wyborców
	klasa $ResultChartView$	wizualizacja wyników wyborów, klasa dziedziczy po klasie $ScatterChartMixin$, pobranie współrzędnych kandydatów, wyborców oraz zwycięzców

3.4. Moduł generacji i wczytywania wyborów

3.4.1. Opis ogólny

Moduł odpowiedzialny za generację wyborów z rozkładu normalnego oraz wczytywanie wyborów z pliku formatu *.soc*. Moduł współpracuje z modułem zarządzania wyborami, któremu zleca po wykonaniu swoich zadań, stworzenie i wysłanie użytkownikowi odpowiedniej strony internetowej. Moduł zapewnia wygenerowanie wyborów z rozkładu normalnego według wskazanych parametrów oraz wali-

dację danych przy wczytywaniu wyborów z pliku. Po stworzeniu wyborów moduł komunikuje się z bazą danych w celu utrwalenia wyborów.

3.4.2. Komponenty programowe

Typ komponentu	Komponenty	Wykorzystanie
Widoki	klasa $ElectionLoadDataFormView$	wczytanie wyborów z pliku
	klasa $ElectionGenerateDataFormView$	generacja wyborów z rozkładu normalnego
Szablony	$election_load_data.html$	strona z formularzem do wskazania pliku
	$election_generate_data.html$	strona z formularzem do wskazania parametrów wyborów i rozkładu normalnego
	$election_details.html$	strona wyświetlana po poprawnym wczytaniu danych z pliku
Formularze	klasa $ElectionLoadDataForm$	wczytanie z pliku

3.5. Moduł obliczania wyników wyborów

3.5.1. Opis ogólny

Moduł odpowiedzialny za obliczanie wyników wyborów. Zapewnia różne algorytmy do wykonania zadania. Użytkownik ma wybór między algorytmem genetycznym, dwoma algorytmami zachłannymi oraz algorytmem typu brute-force. Moduł współpracuje z modułem zarządzania wyborami, który zleca mu wykonanie zadania.

3.5.2. Komponenty programowe

Wszystkie komponenty programowe dotyczące modułu obliczania wyników wyborów zawierają się w pakiecie $ecs.elections.algorithms$.
Klasy odpowiedzialne za poszczególne algorytmy:

- $BruteForce$ - odpowiedzialna za algorytm typu brute-force
- $GreedyAlgorithm$ - odpowiedzialna za algorytm zachłanny zależny od parametru p
- $GreedyCC$ - odpowiedzialna za algorytm zachłanny niezależny od parametru p
- $GeneticAlgorithm$ - odpowiedzialna za algorytm genetyczny

3.6. Moduł URL Resolver

3.6.1. Opis ogólny

Moduł odpowiedzialny za przekazywanie żądań otrzymywanych przez klienta do odpowiednich modułów.

3.6.2. Komponenty programowe

Przyporządkowania żądań użytkownika do odpowiednich modułów (adresów URL do widoków) znajdują się w pliku $ecs.elections.urls.py$.