Autor: Przemysław Rewiś

Tytuł: Aplikacja do realizacji badań poparcia dla partii politycznych

Cel projektu:

Celem projektu jest stworzenie narzędzia pozwalającego na przeprowadzanie badań opinii publicznej na temat poparcia dla różnych partii politycznych w kraju. Aplikacja będzie umożliwiać zbieranie danych od respondentów w sposób szybki, łatwy i skuteczny, dzięki czemu wyniki badań będą mogły być wykorzystane do analizy.

Funkcjonalności aplikacji:

- podział użytkowników na respondentów i administratorów
- samodzielna rejestracja respondentów
- logowanie
- dodawanie nowych administratorów przez administratora
- respondent będzie mógł zapoznać się z informacjami o partiach
- administrator będzie mógł edytować dane partii oraz dodawać je i usuwać
- aplikacja umożliwi przeprowadzenie badań popracia wśród respondentów
- aplikacja będzie umożliwiała analizę wyników badań administratorowi poprzez prezentację wyników
- aplikacja będzie integrować się z bazą danych, umożliwiając automatyczne zapisywanie wyników badań i ich przetwarzanie.

Rozwiązania technologiczne:

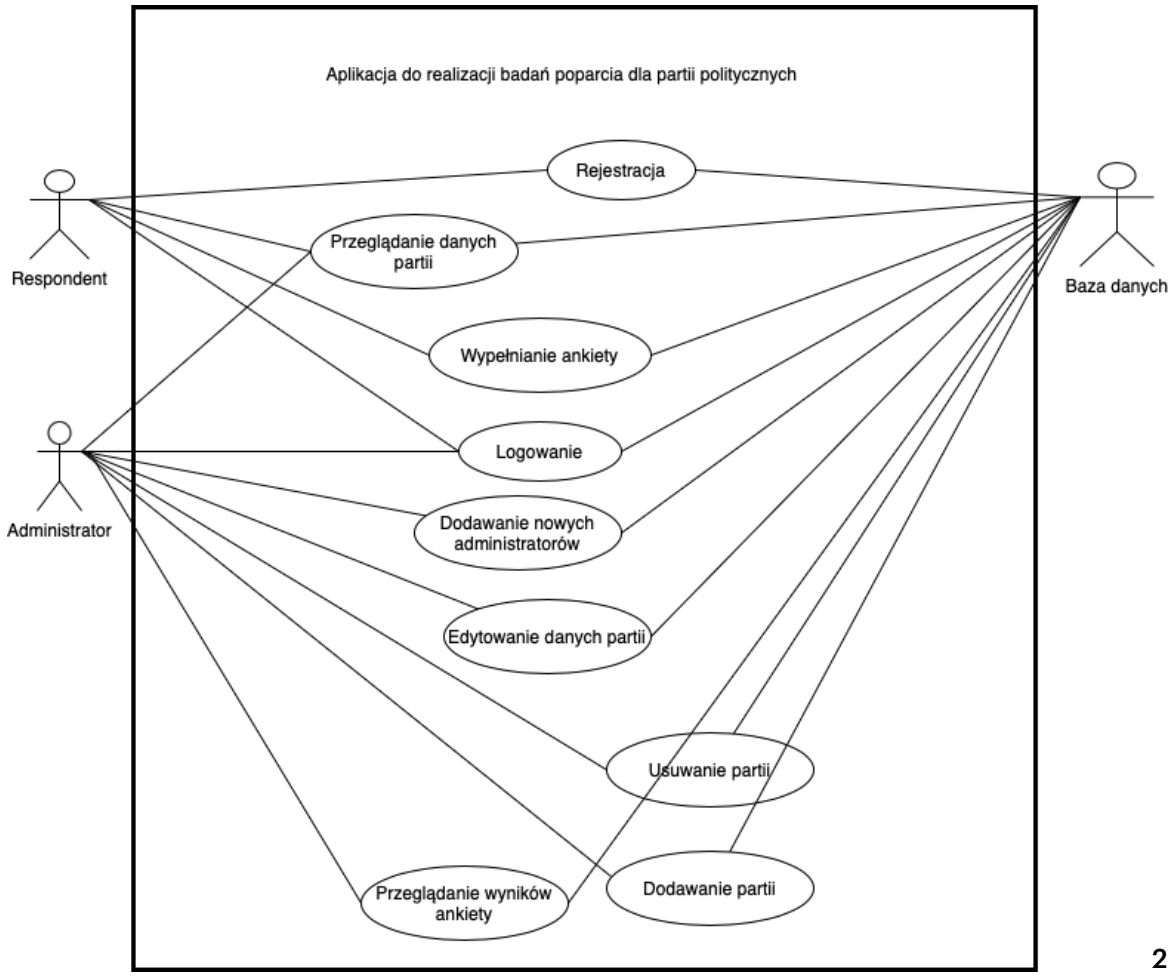
- aplikacja stworzona zostanie z wykorzystaniem języka Java
- będzie ona oparta na architekturze klient-serwer,
- do budowania serwera zostanie wykorzystany framework Spring Boot,
- do budowania klienta zostaną wykorzystane frameworki Vaadin i Spring Boot,
- baza danych PostgreSQL

Przewidywane narzędzia projektowe:

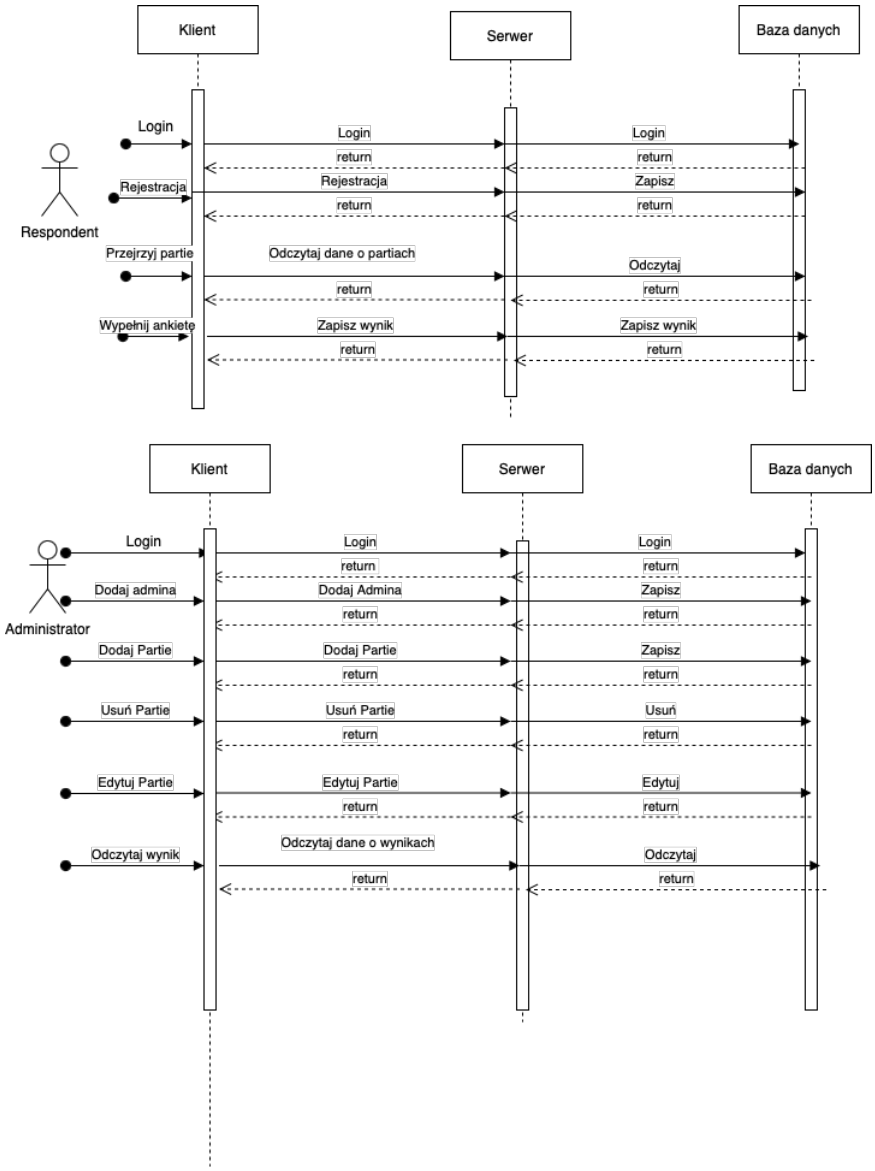
- środowisko programistyczne IntelliJ Idea
- Docker
- ElephantSQL

Diagramy UML:

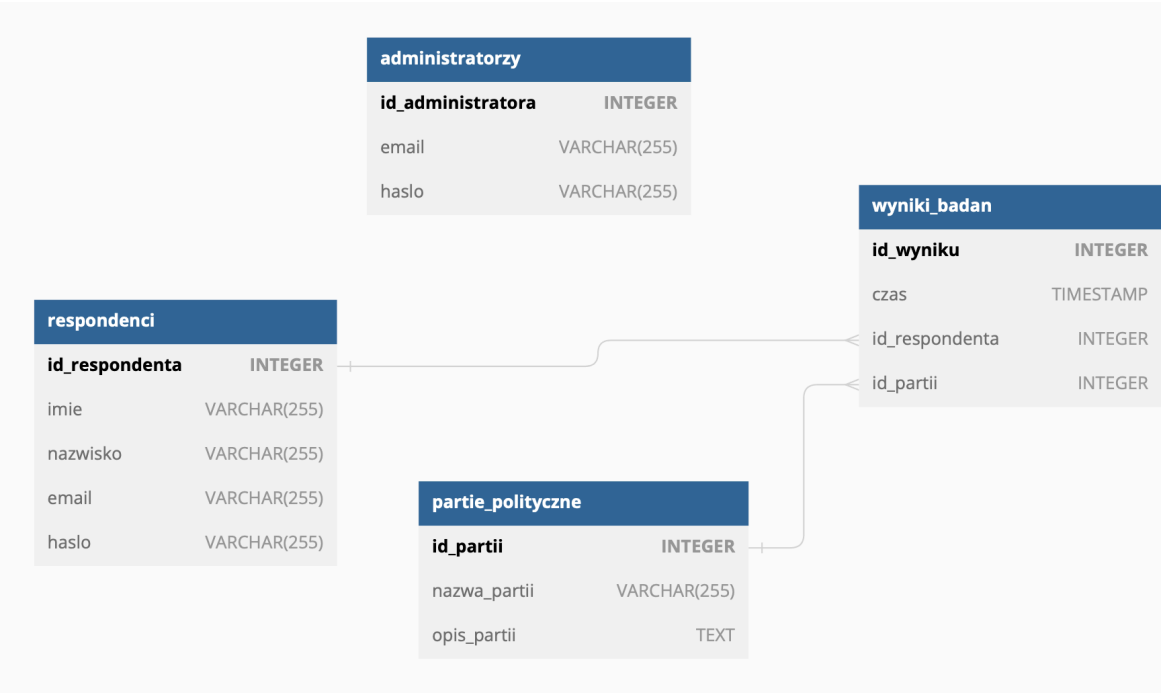
- diagram przypadków użycia



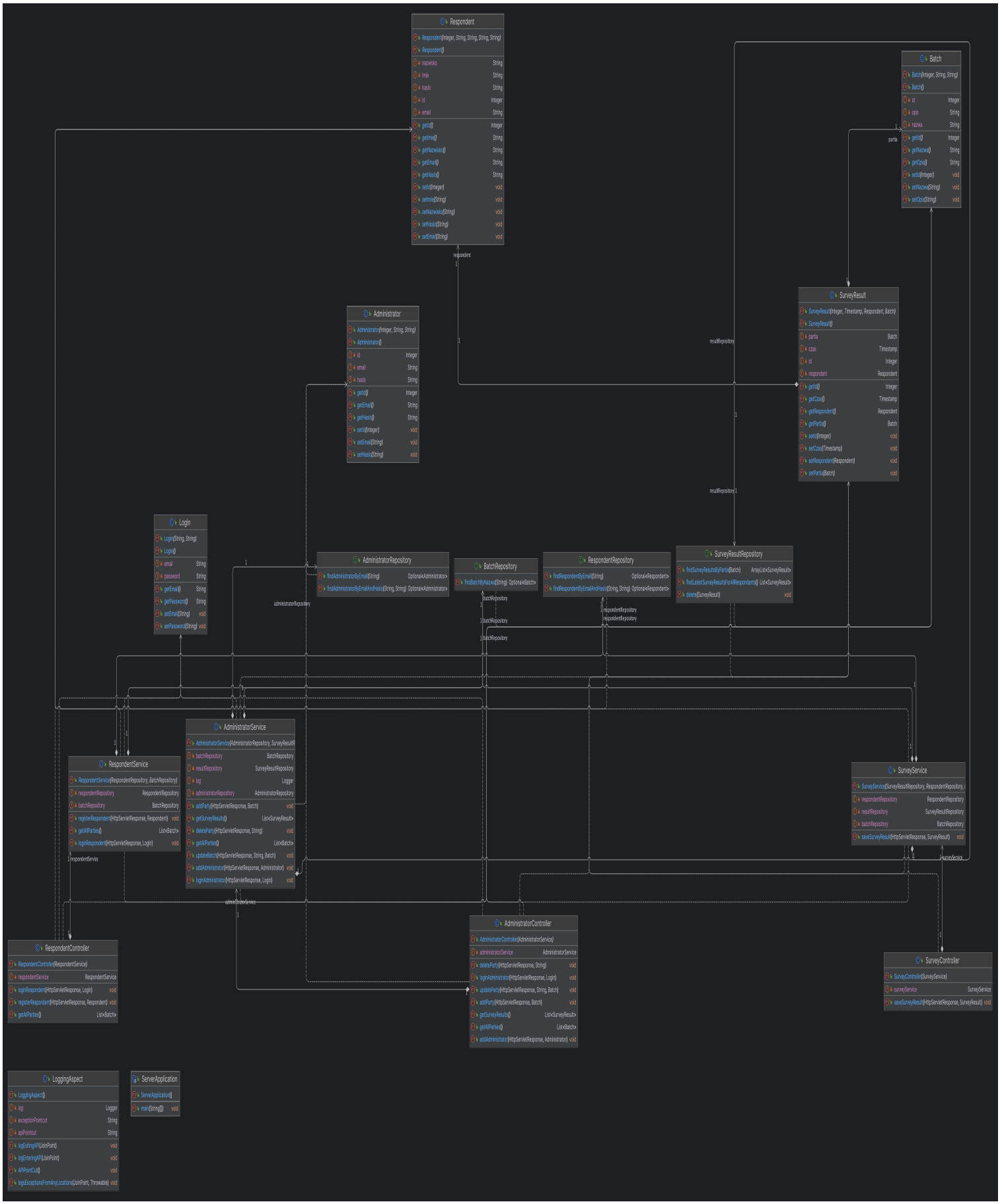
- diagram przebiegu



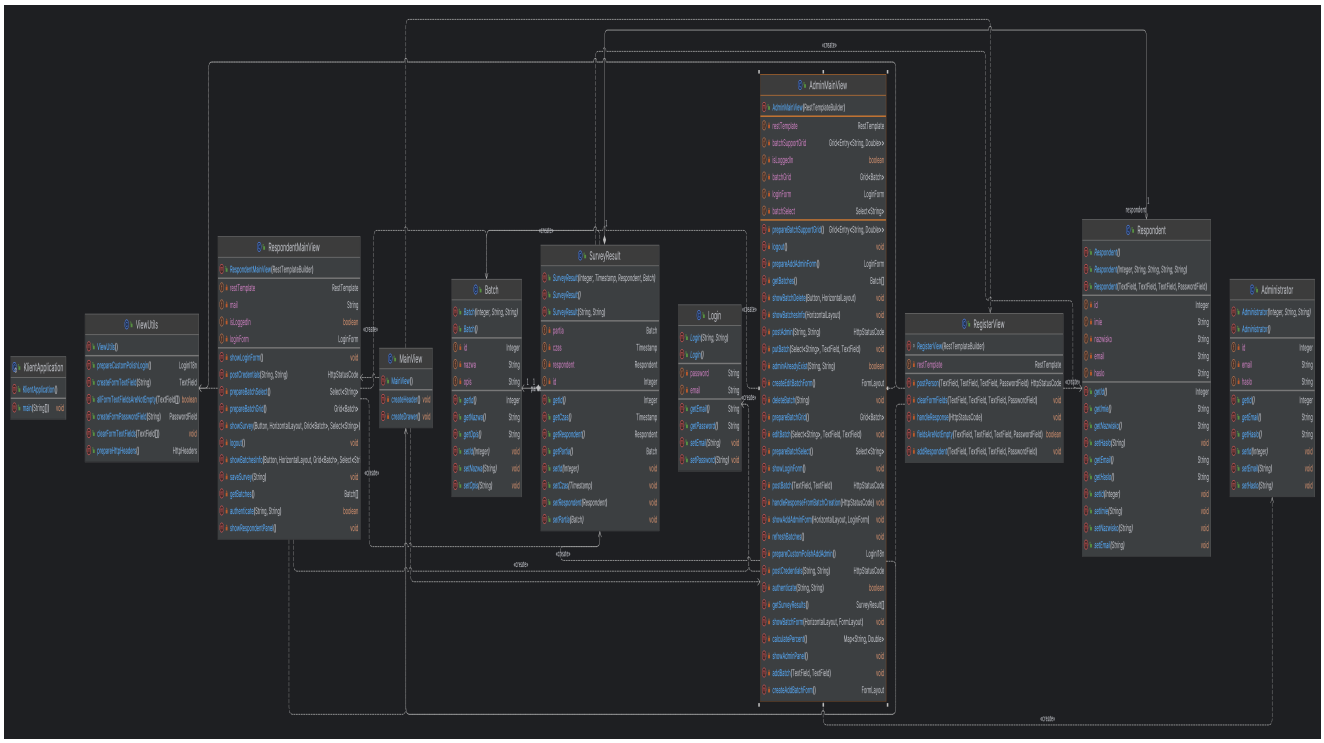
- diagram ERD bazy danych



- diagram klas serwer



- diagram klas klient



Testy:

Testy serwera są częścią testowej klasy `ServerApplicationTests` i służą do sprawdzenia poprawności działania różnych komponentów aplikacji serwerowej. Krótkie omówienie poszczególnych testów:

1. `contextLoads()`: Ten test sprawdza, czy aplikacja jest w stanie poprawnie załadować kontekst Springa. Jest to test wstępny, który zapewnia, że aplikacja może poprawnie uruchomić się w kontenerze Springa.

2. `controllersLoads()`: Test ten sprawdza, czy kontrolery zostały poprawnie załadowane i wstrzyknięte przez Springa. Wykorzystując asercje z biblioteki AssertJ, sprawdzane jest, czy kontrolery `administratorController`, `respondentController` i `surveyController` nie są nullami, co oznacza poprawne ich załadowanie.

3. `servicesLoads()`: Ten test sprawdza, czy serwisy zostały poprawnie załadowane i wstrzyknięte przez Springa. Podobnie jak w poprzednim teście, wykorzystywane są asercje, aby sprawdzić, czy `administratorService`, `respondentService` i `surveyService` nie są nullami.

4. `aspectsLoads()`: Test ten sprawdza, czy aspekt `LoggingAspect` został poprawnie załadowany i wstrzyknięty przez Springa. Przy użyciu asercji sprawdzane jest, czy `myAspect` nie jest nullem.

Testy klienta są częścią testowej klasy `KlientApplicationTests` i służą do sprawdzenia poprawności działania różnych komponentów aplikacji frontendowej. Krótkie omówienie poszczególnych testów:

1. `contextLoads()`: Ten test sprawdza, czy aplikacja jest w stanie poprawnie załadować kontekst Springa. Jest to test wstępny, który zapewnia, że aplikacja może poprawnie uruchomić się w kontenerze Springa.
2. Wykonane zostały testy manualne, które miały na celu sprawdzenie komunikacji pomiędzy klientem i serwerem oraz poprawne działanie wszystkich widoków.

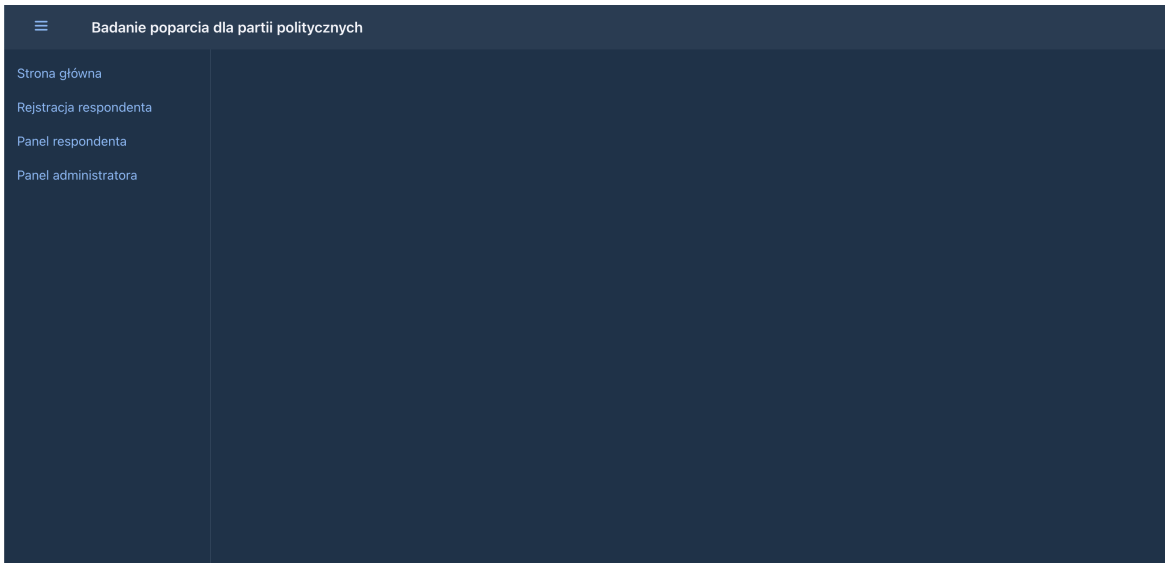
Informacja uruchomieniowa:

Aplikacja do poprawnego działania wymaga, aby najpierw został uruchomimy serwer. Do uruchomienia serwera w postaci kontenera wymagane jest skonfigurowanie Dockera (zalecane pobranie Docker Desktop). Pobieramy obrazy gradle:7.2.0-jdk17 oraz openjdk:17-jdk-slim. Otwieramy terminal w folderze projekt. W terminalu wpisujemy komendę: docker-compose build --no-cache ,a następnie wpisujemy komendę: docker-compose up serwer powinien uruchomić się na porcie 8080. Alternatywnie serwer możemy uruchomić otwierając terminal w folderze projekt/server i

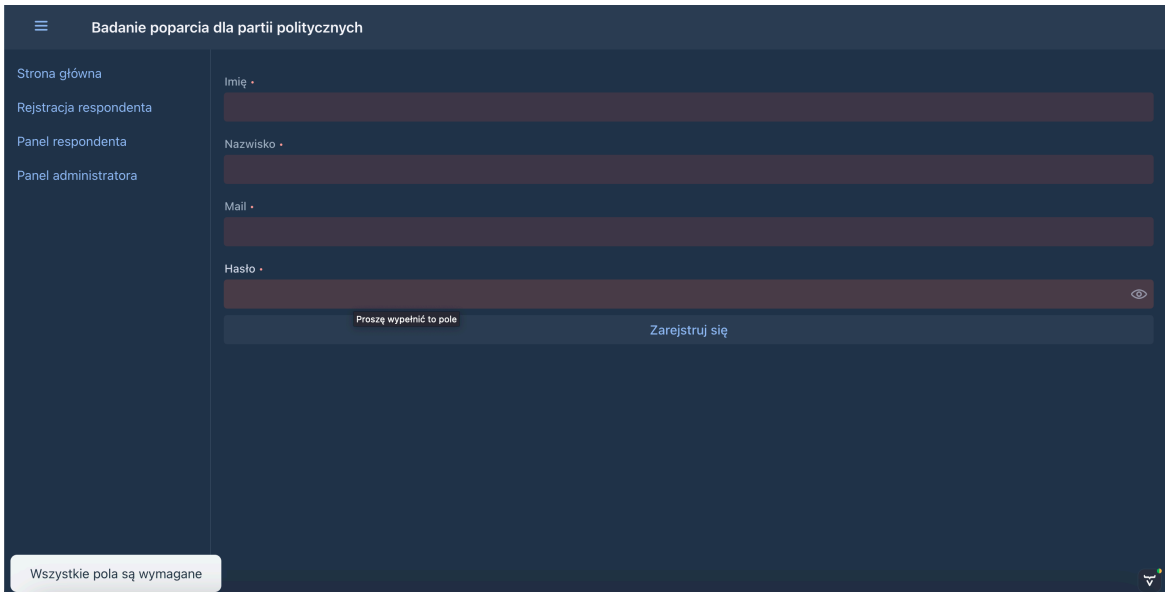
wpisać komendy ./gradlew build i następnie ./gradlew bootRun. Klient uruchamiamy otwierając terminal w folderze projekt/klient i wpisując komendy ./gradlew build i następnie ./gradlew bootRun. Otwieramy link: http://localhost:8081/

Podstawowy podręcznik użytkownika:

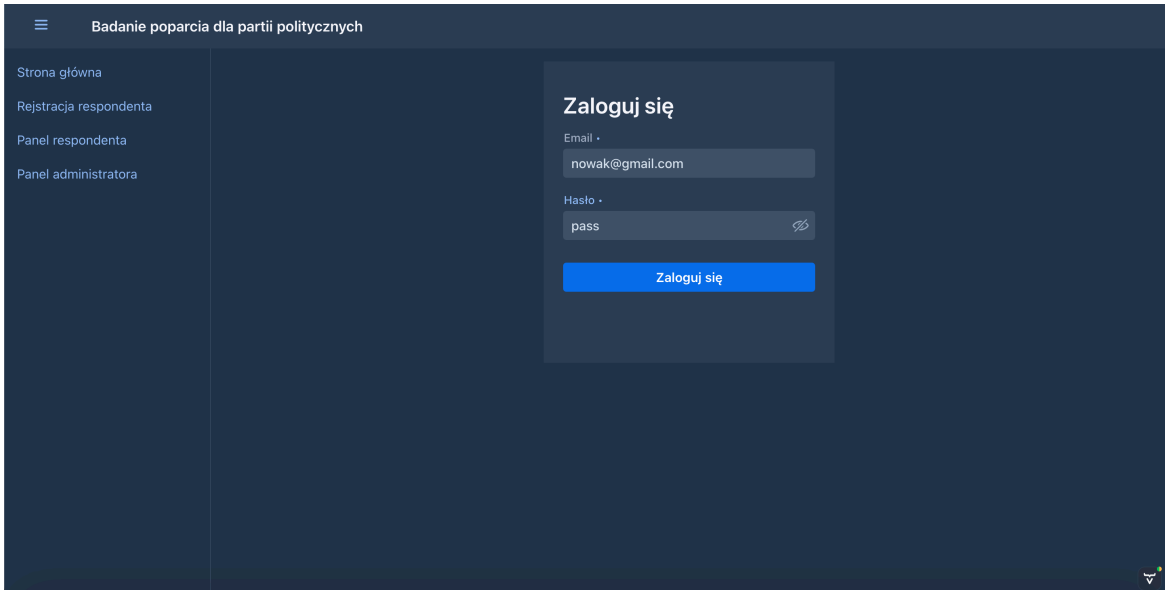
Po wejściu w powyższy link ukaże nam się strona główna. Po lewej stronie mamy menu które służy do poruszania się po aplikacji.



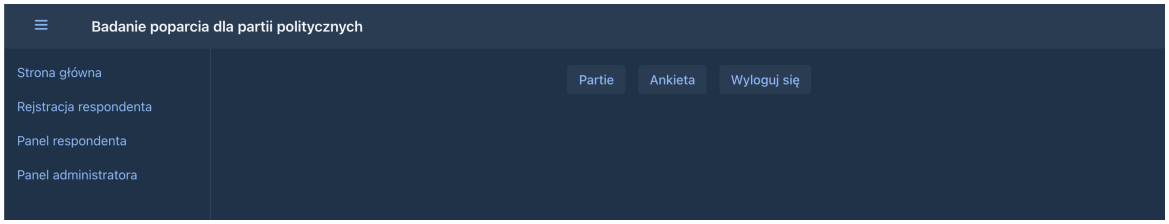
Klikając w napis *Rejestracja respondenta* ukaże nam się formularz, który umożliwia stworzenie nowego konta respondenta.



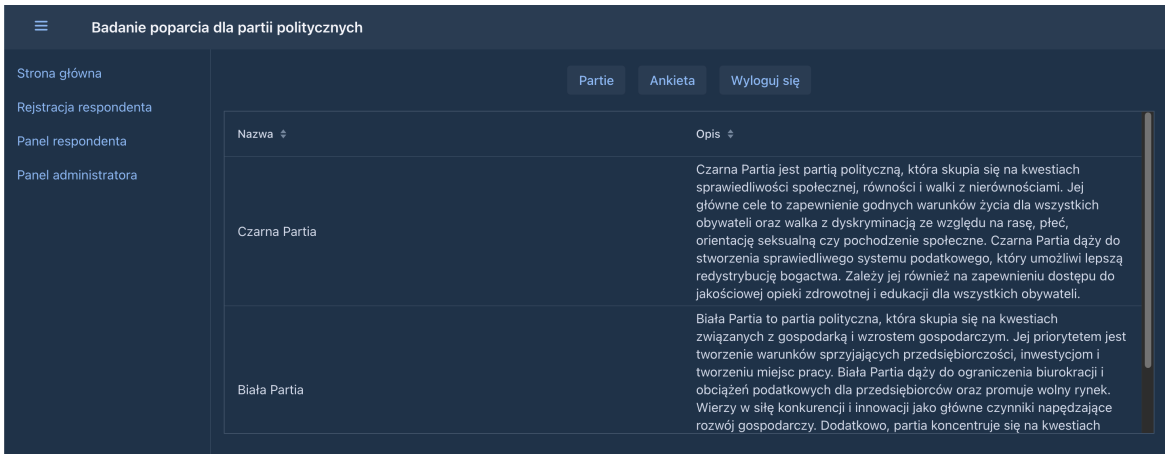
Formularz wymaga wypełnienia wszystkich pól jeżeli nie są one uzupełnione aplikacja wyświetla powiadomienie: „Wszystkie pola są wymagane”. Jeżeli osoba podczas rejestracji poda taki sam mail jak inny respondent to aplikacja wyświetla powiadomienie: „Respondent o takim adresie e-mail istnieje”. Jeżeli rejestracja powiodła się aplikacja wyświetla powiadomienie: "Respondent utworzony poprawnie”. Klikając w napis *Panel respondenta* ukaże nam się formularz, który umożliwia zalogowanie się. Przykładowe dane logowania:



Po zalogowaniu ukazuje nam się panel respondenta. Po naciśnięciu przycisku *Wyloguj się* aplikacja ponownie wyświetla formularz logowania.

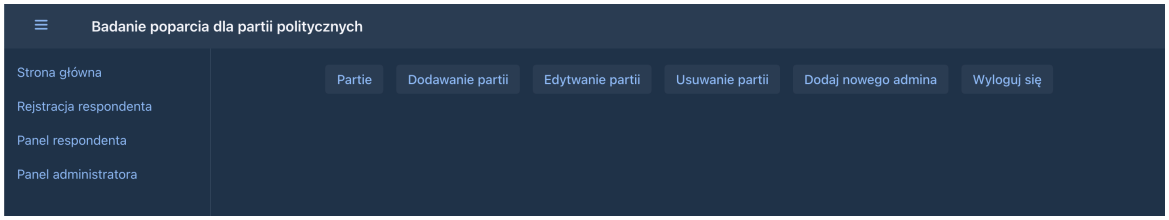


Po naciśnięciu przycisku *Partie* ukażą nam się informacje o partiach.

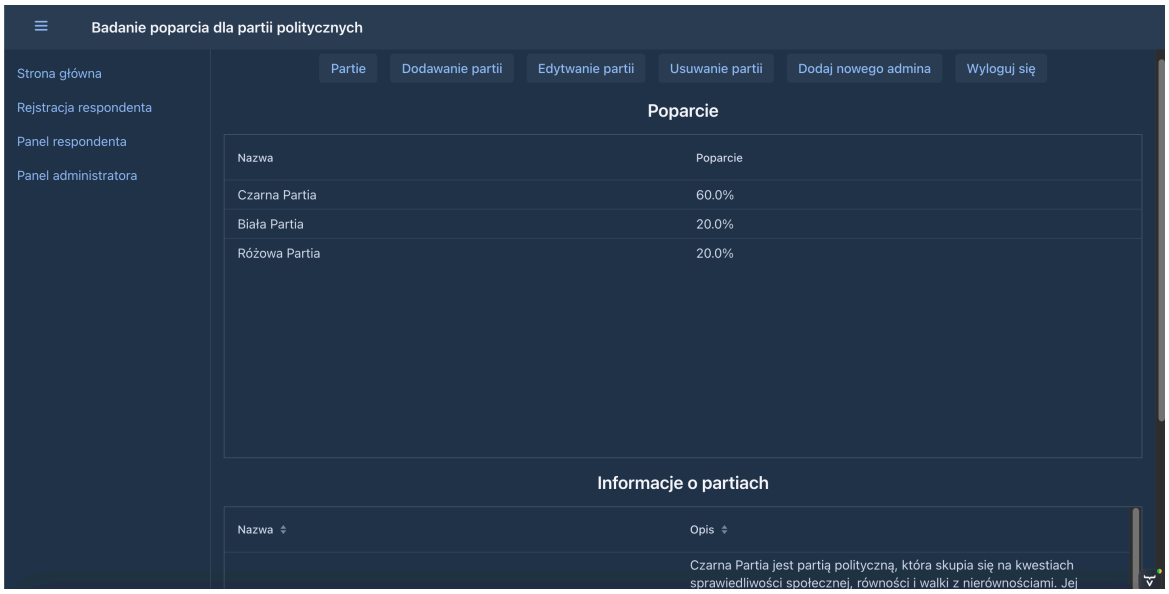


Po naciśnięciu przycisku *Ankieta* możemy wybrać partię, którą popieramy oraz zapisać wybór poprzez kliknięcie przycisku *Zapisz ankiete*.

Klikając w napis *Panel administratora* ukaże nam się formularz, który umożliwia zalogowanie się. Dane do logowania Email: admin@gmail.com Hasło: pass. Po zalogowaniu ukazuje nam się panel administratora.



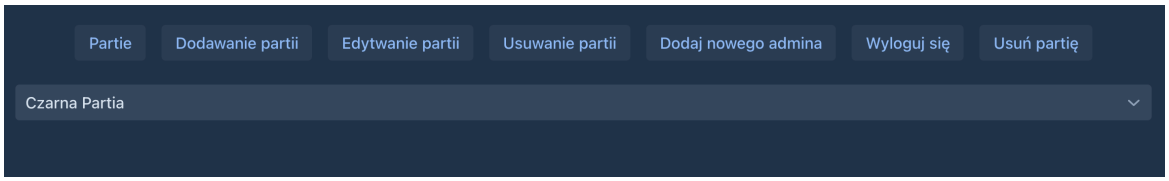
Po naciśnięciu przycisku *Partie* ukażą nam się informacje o aktualnym poparciu oraz o partiach.



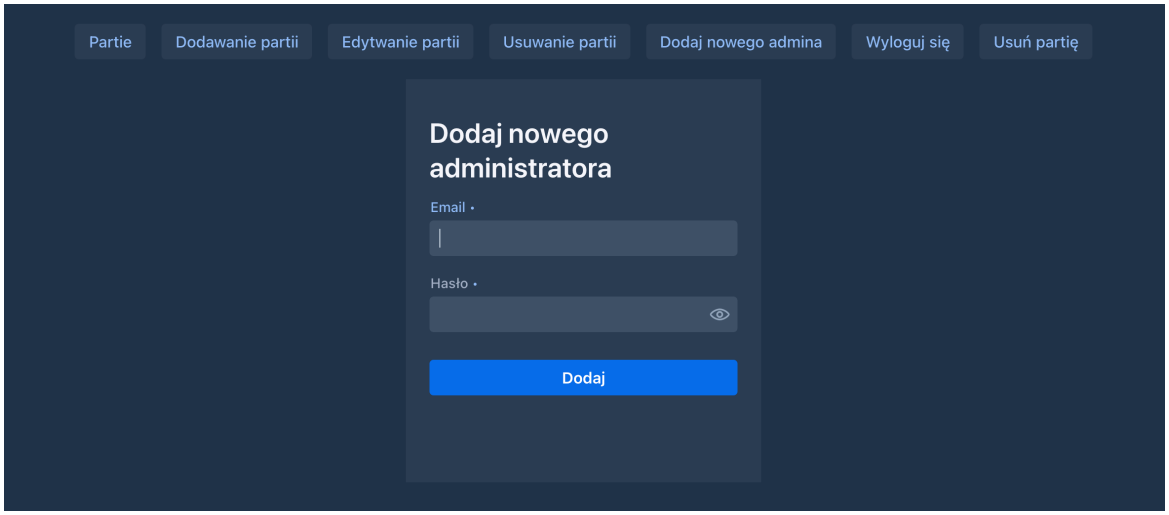
Po naciśnięciu przycisku *Dodawanie partii* ukaże nam się poniższy formularz

Po naciśnięciu przycisku *Edytowanie partii* ukaże nam się poniższy formularz, w którym wybieramy partię do edycji, a następnie wprowadzamy do niego nowe dane.

Po naciśnięciu przycisku *Usuwanie partii* ukaże nam się poniższy formularz. Wybieramy w nim partię, a następnie klikamy przycisk Usuń partię, aby ją skasować.



Po naciśnięciu przycisku *Dodaj nowego admina* ukaże nam się poniższy formularz.



Po naciśnięciu przycisku *Wyloguj się* aplikacja ponownie wyświetla formularz logowania. Wszystkie omówione wcześniej formularze sygnalizują użytkownikowi poprzez powiadomienia czy dana operacja zakończyła się sukcesem lub porażką.