

Sławomir Kryszтовиak  
Marek Nowicki  
Bartosz Paprzycki  
Jakub Prabucki  
Łukasz Siewierski

# ZawodyWEB

## Specyfikacja Wymagań Systemu

# Spis treści

<b>Rozdział 1. Opis produktu</b>	2
1.1. Cel produktu	2
1.2. Grupa docelowa	2
1.3. Wymagania sprzętowe i programowe	3
1.4. Bezpieczeństwo	3
<b>Rozdział 2. Baza</b>	4
<b>Rozdział 3. Cechy produktu</b>	5
<b>Rozdział 4. Przypadki użycia</b>	7
4.1. Wysyłanie rozwiązania	7
4.2. Sprawdzanie rozwiązania przez Judge	7
4.3. Rejestracja użytkownika	7
4.4. Logowanie użytkownika	8
4.5. Dodawanie zawodów, serii, zadań i testów	8
4.6. Zmiana profilu/hasła użytkownika przez administratora	8
4.7. Wysyłanie pytań do treści zadania	9
4.8. Dodawanie odpowiedzi na pytania	9
4.9. Kopiowanie zadania do innych zawodów/serii	9
4.10. Ręczne ocenianie rozwiązania	9
<b>Rozdział 5. Prototyp graficzny</b>	10

## Rozdział 1

# Opis produktu

ZawodyWEB jest rozbudowaną aplikacją, napisaną głównie w języku Java, służącą do przeprowadzania różnego rodzaju zawodów wykorzystując jedną instancję aplikacji. Ponadto pozwala na jednoczesne przeprowadzenie kilku zawodów, dzięki czemu infrastruktura komputerowo-sieciowa może być używana optymalnie.

Aplikacja składa się z dwóch części:

- interfejsu WWW;
- infrastruktury sprawdzającej rozwiązania.

Dzięki przejrzystemu i funkcjonalnemu interfejsowi WWW aplikacja może poszczycić się prostotą użytkowania, więc nawet niezbyt doświadczeni użytkownicy komputera mogą z niej korzystać.

### 1.1. Cel produktu

ZawodyWEB służy do przeprowadzania konkursów programistycznych następujących typów:

- ACM, gdzie główny nacisk kładzie się na szybkie i bezbłędne rozwiązanie jak największej liczby zadań, a wynik działania programu jest widoczny dla zawodników niemal natychmiast;
- PA, gdzie każde zadanie ma różne testy, sprawdzające poprawność rozwiązania i czas działania, a wynik działania programu dla różnych testów dostępny jest dopiero po zakończeniu przesyłania rozwiązań (przy czym ilość zgłoszeń nadesłanych przez zawodnika na każde zadanie może być ograniczona);
- KI, gdzie zawodnicy mogą przysyłać swoje rozwiązania, które następnie są dostępne poprzez WWW: zawodnicy mogą przetestować swoje rozwiązania dokładnie na takim samym systemie na jakim będą oceniali je sędziowie.

Poza wymienionymi wyżej typami konkursów programistycznych, możliwe jest dostosowanie systemu, do przeprowadzania sprawdzianów czy nauki, choć w dalszej części dokumentacji skupimy się głównie na konkursach. Prostota użytkowania aplikacji pozwala skoncentrować się na merytorycznej części konkursów a nie na aspektach technicznych.

### 1.2. Grupa docelowa

Użytkowników projektu możemy podzielić na dwie grupy:

- *Koordynatorzy* – wszelkie osoby chcące w zautomatyzowany sposób publikować problemy o charakterze informatycznym, a następnie sprawdzać poprawność nadsyłanych rozwiązań; w szczególności *organizatorzy* konkursów informatycznych, *sędziowie* sprawdzający rozwiązania zawodników oraz *trenerzy* zespołów turniejowych.
- *Zawodnicy* – wszelkie osoby nadsyłające rozwiązania problemów publikowanych w systemie przez *Koordynatorów* w celu ich automatycznej oceny; głównie uczestnicy konkursów lub treningów informatycznych.

### 1.3. Wymagania sprzętowe i programowe

Do uruchomienia systemu wymagany jest co najmniej jeden komputer klasy PC, posiadający system operacyjny oraz zasoby wystarczające do uruchomienia:

- Java SE Runtime Environment 6
- PostgreSQL
- Tomcat 6

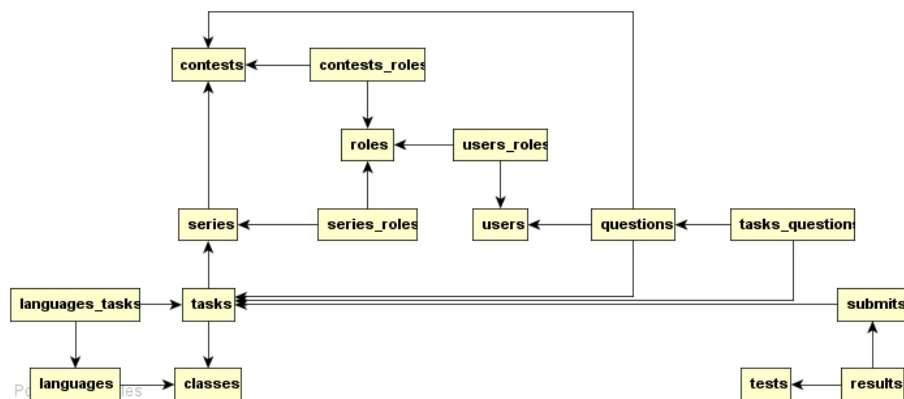
### 1.4. Bezpieczeństwo

Dzięki systemowi kopii zapasowych oraz dobrze skonfigurowanym politykom bezpieczeństwa, aplikacja zapewnia odpowiedni poziom zabezpieczeń w razie awarii oraz przed włamaniami, co jest pożądane przy organizacji konkursów informatycznych.

## Rozdział 2

# Baza

Serwerem bazodanowym wykorzystywanym przez aplikację jest PostgreSQL. Poniższy rysunek przedstawia uproszczony schemat bazy danych.



Rysunek 2.1. Schemat bazy danych

## Rozdział 3

# Cechy produktu

Poniżej znajdują się planowane cechy produktu ułożone według priorytetów.

Cecha	Ważność	Trudność	Ryzyko
Czytelne nazewnictwo kolumn	10	1	1
Dymki podpowiedzi w rankingu po najechaniu na nazwę zadania	10	2	1
Możliwość dodawania zawodów	10	3	1
Możliwość edycji zawodów	10	3	1
Możliwość usuwania zawodów	10	3	1
Możliwość dodawania tur/serii	10	3	1
Możliwość edycji tur/serii	10	3	1
Możliwość usuwania tur/serii	10	3	1
Możliwość dodawania zadań	10	3	1
Możliwość edycji zadań	10	3	1
Możliwość usuwania zadań	10	3	1
Możliwość czytania zadań	10	3	1
Możliwość zapisywania się na zawody	10	3	1
Możliwość wysyłania rozwiązań	10	4	1
Możliwość ręcznego sprawdzania zadań w języku php, java	10	4	1
Baza na postgresie	10	4	1
Punkty za zadania	10	4	1
Typ rankingu	10	4	1
Typ zawodów	10	4	1
Role dla użytkowników	10	6	1
Przejrzysta obsługa przetwarzania zgłoszenia	10	10	1
Możliwość szybkiego wysyłania rozwiązań	9	2	1
Wysyłanie rozwiązań z pola tekstowego	9	2	1
Wysyłanie rozwiązań z pliku	9	2	1
Wyświetlanie czasu od rozpoczęcia tury	9	2	1
Możliwość dodawania testów	9	3	1
Możliwość edycji testów	9	3	1
Możliwość usuwania testów	9	3	1
Możliwość ustalania kolejności testów	9	3	1
Możliwość ustalania kolejności zadań	9	3	1
Bezpośrednie linki do zadań	9	4	1
Możliwość zarządzania użytkownikami	9	6	1
Możliwość sprawdzania ręcznego zadań	8	2	1
Możliwość ustalania kolejności testów	8	2	1
Możliwość blokady zbyt częstych zgłoszeń	8	3	1
Możliwość łatwego obejrzenia punktów, źródła i zadania w rozwiązaniach	8	4	1
Ranking do zawodów	8	5	1
Ranking do tury	8	5	1
Ranking za zadania	8	5	1
Wybór sposobu rejestracji na zawody	8	5	1
Ładna ikonka do zakładek (favicon)	8	6	1
Możliwość dodawania języków	8	6	2

Możliwość usuwania języków	8	6	2
Możliwość edycji języków	8	6	2
Przyjazny interfejs	8	8	1
Ustalenie maksymalnej ilości zgłoszeń	7	3	1
System pytań i odpowiedzi	7	4	1
Opisy błędów wykonania, kompilacji, itd.	7	4	1
Wysyłanie e-mailem rozwiązania	7	4	1
Możliwość blokady użytkowników	7	5	1
Podział na zawody zamknięte i otwarte	7	5	1
Możliwość wyboru sposobu definiowania użytkowników na zawody zamknięte	7	5	1
Możliwość drukowania kodu	7	8	1
Restfull links	6	2	1
Kolorowanie składni	6	3	1
Captcha przy rejestracji	6	3	1
Równoległe sprawdzanie	6	3	1
Podział zawodów na aktualne i zakończone	6	4	1
Branie pod uwagę pierwszego (pod względem czasu) rozwiązania, które uzyskało maksymalną ilość punktów	6	5	1
Możliwość wyboru zawodów	6	5	1
Przycisk startujący zawody	5	1	1
Przyjazny wygląd drukowanego kodu (css)	5	2	1
Możliwość wyboru zawodów przy rejestracji użytkownika	5	4	1
Możliwość zadawania pytań	4	2	1
Lista użytkowników danych zawodów	4	2	1
Możliwość odpowiadania na pytania	4	3	1
System przypominania hasła	4	3	1
Filtrowania rozwiązań wg. „zadanie” i „użytkownik”	4	5	1
Możliwość dodawania zadań w formacie pdf	3	2	1
Udostępnianie kodów rozwiązań (różne sposoby)	3	4	1
Ukrywanie specjalnych użytkowników w rankingu	3	5	1
Możliwość ukrywania menu	2	2	1
Wyświetlanie informacji o dacie rejestracji i ostatniej aktywności w serwisie danego użytkownika	2	2	1
Możliwość sprawdzenia rankingu o zadanej dacie	2	3	1
Rozsądne drukowanie strony (podobne do Wikipedii)	1	2	1
Tagowanie zadań	1	3	1
Logowanie aktywności na stronie	1	7	1
Obsługa zadań optymalizacyjnych	1	7	1

## Rozdział 4

# Przypadki użycia

### 4.1. Wysyłanie rozwiązania

1. Użytkownik wysyła rozwiązanie do serwera WWW
2. Serwer WWW zapisuje informacje o przesłanym rozwiązaniu do bazy danych oraz kolejkuje rozwiązanie jako niesprawdzone
3. Serwer WWW przesyła dane o rozwiązaniu do JudgeManager
4. JudgeManager kolejkuje rozwiązanie jako niesprawdzone oraz przesyła dane o rozwiązaniu do jednego z Judge
5. Judge komunikuje się z bazą danych aby pobrać dane testowe oraz sprawdza rozwiązanie
6. Judge informuje JudgeManagera o tym, że rozwiązanie zostało sprawdzone i przesyła wynik testu do bazy danych
7. Judge Manager informuje serwer WWW, że sprawdzanie zostało zakończone oraz usuwa ze swojej kolejki rozwiązanie
8. Serwer WWW usuwa z kolejki rozwiązanie

### 4.2. Sprawdzanie rozwiązania przez Judge

1. Judge dostaje identyfikator rozwiązania od JudgeManagera
2. Judge odczytuje z bazy danych informacje o rozwiązaniu (nazwa pliku, kod źródłowy, język, itp)
3. Judge odczytuje z bazy danych informacje o języku (nazwy klas potrzebne do kompilacji, uruchamiania oraz porównywania wyników)
4. Judge kompiluje rozwiązanie użytkownika
  - a) Jeśli nie powiedzie się kompilacja Judge zwraca wynik i kończy działanie
5. Judge zaczyna sprawdzać poprawność rozwiązania. Dla każdego testu wykonuje poniższe czynności:
  - a) Pobiera dane wejściowe
  - b) Uruchamia za pomocą klasy, pobranej z bazy, program na danych wejściowych i zapamiętuje wynik
  - c) Porównuje wynik z prawidłową odpowiedzią używając specjalnej klasy
  - d) Zapamiętuje wynik
6. Judge wysyła informację do JudgeManagera o tym, że rozwiązanie zostało sprawdzone
7. Judge zapisuje wynik do bazy danych

### 4.3. Rejestracja użytkownika

1. Użytkownik wypełnia formularz podając
  - a) login
  - b) hasło
  - c) adres e-mail
  - d) imię
  - e) nazwisko
  - f) datę urodzenia
  - g) adres zamieszkania
  - h) informacje o szkole



- i) informacje o nauczycielu
2. Przepisuje token z obrazka (CAPTCHA)
3. Naciska przycisk *Rejestruj*
  - a) Jeśli dany login jest już zajęty proszony jest o ponowne wypełnienie tego pola formularza
4. Otrzymuje potwierdzenie rejestracji na podany adres e-mail

#### 4.4. Logowanie użytkownika

1. Użytkownik wprowadza następujące dane:
  - a) login
  - b) hasło
2. Naciska przycisk *Zaloguj*
  - a) Jeśli użytkownik podał nieprawidłowe dane, to zostanie o tym poinformowany

#### 4.5. Dodawanie zawodów, serii, zadań i testów

Dodawać zawody, serie, zadania i testy może jedynie osoba uprawniona, wykorzystując panel administracyjny.

1. Uprawniony użytkownik wprowadza następujące dane:
  - a) W przypadku dodawania zawodów:
    - i. Nazwę konkursu
    - ii. Typ konkursu
    - iii. Datę rozpoczęcia
    - iv. Informacje techniczne oraz zasady
    - v. Widoczność
  - b) W przypadku dodawania serii:
    - i. Zawody, do których ma należeć dodawana seria
    - ii. Nazwę serii
    - iii. Daty rozpoczęcia i zakończenia konkursu oraz *zamrożenia* rankingu
    - iv. Wysokość kary doliczanej za błędne rozwiązanie
  - c) W przypadku dodawania zadań:
    - i. Serię, do której ma należeć dodawane zadania
    - ii. Nazwę zadania oraz jej skrót
    - iii. Treść zadania, w postaci tekstu oraz pliku PDF
    - iv. Limit pamięci
  - d) W przypadku dodawania testu:
    - i. Zadanie, do którego ma należeć dodawany test
    - ii. Dane wejściowe oraz wyjściowe (także skompresowane)
    - iii. Limit czasu
    - iv. Liczbę punktów
    - v. Widoczność

#### 4.6. Zmiana profilu/hasła użytkownika przez administratora

Administrator posiada możliwość edycji dowolnych atrybutów użytkowników. Aby tego dokonać:

1. Wyszukuje użytkownika na liście (np. za pomocą loginu, imienia lub nazwiska)
2. Zmienia pola formularza
3. Naciska przycisk *Zapisz*

#### 4.7. Wysyłanie pytań do treści zadania

Uczestnik konkursu ma możliwość stawiania pytań dotyczących treści zadania lub pytań ogólnych. Aby wysłać pytanie uczestnik wykonuje następujące czynności:

1. Wybiera z rozwijalnej listy nazwę zadania lub pozycję *Ogólne*
2. Wypełnia treść pytania
3. Naciska przycisk *Wyślij*

#### 4.8. Dodawanie odpowiedzi na pytania

Użytkownicy uprawnieni do odpowiadania na pytania mogą to uczynić korzystając z panelu administracyjnego. Aby odpowiedzieć na pytanie użytkownik wykonuje następujące czynności:

1. Wskazuje w tabeli pytanie, na które zamierza odpowiedzieć, lub tworzy nowe puste pytanie i wybiera zadanie, którego ono dotyczy
2. W formularzu odpowiedzi
  - a) Odpisuje na zadane pytanie
  - b) Decyduje, czy odpowiedź ma być widoczna tylko dla zadającego pytanie, czy dla wszystkich uczestników konkursu
  - c) Naciska przycisk *Odpowiedz*

#### 4.9. Kopiowanie zadania do innych zawodów/serii

1. Użytkownik uprawniony wybiera zadanie do skopiowania oraz zawody i serię docelową
2. Po stronie serwera kopiowane są wszystkie dane o zadaniu (treść, limity, testy, itp) oraz rozwiązania użytkowników; kopiowane jest rozwiązanie użytkownika, w którym otrzymał on największą liczbę punktów. Jeśli użytkownik otrzymał zero punktów, jego rozwiązanie jest ignorowane.

#### 4.10. Ręczne ocenianie rozwiązania

1. Użytkownik uprawniony wybiera z tabeli anonimowe rozwiązania. Ma możliwość:
  - a) Przeglądania kodu źródłowego
  - b) Pobierania kodu źródłowego
  - c) Testowania rozwiązań na serwerze (w przypadku zadań wymagających interakcji)
  - d) Testowania pobranych rozwiązań na własnym komputerze
  - e) Przeglądania i pobierania (także wersji skompresowanej) testów powiązanych z zadaniem
  - f) Oceniania rozwiązań poszczególnych testów (punkty, czas działania, komentarz)
  - g) Komentowania całego rozwiązania

## Rozdział 5

# Prototyp graficzny

Wydział Matematyki i Informatyki  
Uniwersytetu Mikołaja Kopernika w Toruniu

2008-12-15 22:56:48

**ZawodyWEB - Eliminacje WMiI UMK do AMPPZ 2008**

Zawody Mój profil Wyślij teraz Zaloguj O zawodach

**Logowanie**

Login:

Hasło:

Zapamiętaj login: ☐

Zaloguj

Rysunek 5.1. Ekran logowania

Wydział Matematyki i Informatyki  
Uniwersytetu Mikołaja Kopernika w Toruniu

2008-12-15 22:56:48

**ZawodyWEB - Eliminacje WMiI UMK do AMPPZ 2008**

Zawody Mój profil Wyślij teraz Zaloguj O zawodach

**Mój profil**

Imię:

Nazwisko:

Data urodzenia:

Adres zamieszkania:

Szkoła:

Nauczyciel prowadzący:

Gadu-gadu:

E-mail:

Powiadomienia na e-mail: ☐

Nazwa użytkownika:

Hasło:

Rysunek 5.2. Podgląd i edycja profilu



Rysunek 5.3. Wybór zawodów



Rysunek 5.4. Ekran wyboru zadania

The screenshot shows the website interface for the 'ZawodyWEB - Eliminacje WMiI UMK do AMPPZ 2008'. The header includes the logo of the Faculty of Mathematics and Informatics of Nicolaus Copernicus University in Torun, the date '2008-12-15 22:56:48', and the title 'ZawodyWEB - Eliminacje WMiI UMK do AMPPZ 2008'. A navigation bar contains links: 'Zawody', 'Mój profil', 'Wyślij teraz', 'Zaloguj', and 'O zawodach'. On the left, a sidebar menu lists 'Zadania', 'Ranking', 'Pytania', 'Harmonogram', and 'Regulamin'. The main content area is titled 'Zadanie: The 3n + 1 problem' and includes a 'Wstęp' (Introduction) section with a paragraph about problem classification in computer science. Below this is the 'Problem' section, which asks the user to consider a specific algorithm and lists its steps:

1. input n
2. print n
3. if n = 1 then STOP
4. if n is odd then
5. else
6. GOTO 2

Rysunek 5.5. Treść zadania

The screenshot shows the submission interface of the 'ZawodyWEB - Eliminacje WMiI UMK do AMPPZ 2008' website. The header and navigation elements are identical to the previous screenshot. The main content area is titled 'Wyślij' (Submit). It contains a form with the following fields and controls:

- 'Zadanie:' dropdown menu with 'Zadanie A' selected.
- 'Język:' dropdown menu with 'C/C++' selected.
- 'Plik:' text input field with a 'Wybierz...' button next to it.
- 'Wyślij plik' button.
- 'Kod:' text area for entering the solution code.
- 'Wyślij kod' button.

Rysunek 5.6. Ekran wysyłania rozwiązań



Wydział Matematyki i Informatyki  
Uniwersytetu Mikołaja Kopernika w Toruniu

2008-12-15 22:56:48

**ZawodyWEB - Eliminacje WMiI UMK do AMPPZ 2008**

Zawody | Moj profil | Wyslij teraz | Zaloguj | O zawodach

**Pytania**

Zadanie:

Temat:

Typ	Temat	Zadanie	Data
Publ.	Jak to działa?	Zadanie A	16:25 01-05-2009
Prywat.	Czy n>0	Zadanie C	11:24 15-05-2009

Rysunek 5.7. Zadawanie pytań



Wydział Matematyki i Informatyki  
Uniwersytetu Mikołaja Kopernika w Toruniu

2008-12-15 22:56:48

**ZawodyWEB - Eliminacje WMiI UMK do AMPPZ 2008**

Zawody | Moj profil | Wyslij teraz | Zaloguj | O zawodach

**Ranking**

Miejsce	Nazwa Użytkownika	Zadania	Czas
1	Halka	A*, B***, C, D, E**	3:25:25
2	Yukoma	A, C**, D, F	2:21:22
3	Kapotek	C, D, E**	1:05:15
4	Turon	C, D	0:25:05
5	Uosena	C, D	0:45:35
6	Jasem	C	0:10:02

Rysunek 5.8. Ranking