

# *Typing Golf*

*Jan Orlik*

*Patrizia Marciniak*

*Maciej Nowak*

*Arkadiusz Wieczorek*

## 1. Opis projektu

---

Przedmiotem niniejszej dokumentacji jest aplikacja internetowa o nazwie Typing Golf. Projekt został utworzony i zaimplementowany przez grupę studentów Wydziału Matematyki i Informatyki Uniwersytetu Adama Mickiewicza w Poznaniu; Jana Orlika, Patrizię Marciniak, Macieja Nowaka oraz Arkadiusza Wieczorka. Inspiracją jej powstania jest zauważalny brak umiejętności używania podczas pracy z komputerem skrótów klawiszowych. Typing Golf oferuje naukę korzystania z niezależnych od używanego systemu skrótów klawiszowych, poprzez wykonywanie niewielkich zadań polegających na przekształcaniu zadanego tekstu w docelową formę w jak najmniejszej ilości uderzeń klawiszy. Celem powstania jest umożliwienie użytkownikowi zwiększenie produktywności podczas korzystania z klawiatury.

## 2. Główne założenia

---

### 2.1 INNOWACYJNOŚĆ

Aplikacja jest pierwszym produktem na rynku przeznaczonym do nabywania nawyku korzystania ze skrótów klawiatury, niezależnych od systemu z jakiego korzystamy. W przeciwieństwie do innych produktów, położony został nacisk na wyuczenie małego zbioru ruchów, jednak mających zastosowanie w pracy z każdym urządzeniem posiadającym klawiaturę. Innowacyjność naszego produktu opiera się również na zastosowaniu algorytmu BFS (przeszukiwania grafu wszerek) w celu określenia najlepszego możliwego sposobu na rozwiązanie zadania. Rozwiązanie to, niezależne od ludzkich możliwości i ograniczeń, gwarantuje użytkownikowi najlepszy rozwój jaki może osiągnąć poprzez tego typu ćwiczenia w danym czasie.

### 2.2 GŁÓWNY CEL: WZROST PRODUKTYWNOŚCI UŻYTKOWNIKA

Aby użytkownik zainteresował się nowym produktem, musi on dostrzegać wyraźne

efekty jego użytkowania. W przypadku aplikacji Typing Golf zysk jest bezsporny; użytkownik nabywa biegłości w korzystaniu z klawiatury, co bezpośrednio wpływa na zaoszczędzony czas. Cecha ta sprawia, że produkt jest atrakcyjny dla firm i organizacji, w których pracownicy spędzają dużo czasu przy pracy z komputerem, jak i dla użytkowników prywatnych, którzy poprzez interesującą formę mają możliwość rozwinąć swoje umiejętności.

### 2.3 BRAK KOSZTÓW UŻYTKOWANIA DLA UŻYTKOWNIKÓW PRYWATNYCH

Darmowy dostęp do aplikacji dla użytkowników prywatnych daje im możliwość rozwoju, bez ponoszenia żadnych kosztów.

### 2.4 REDUKCJA KOSZTÓW UŻYTKOWANIA DLA UŻYTKOWNIKÓW KOMERCYJNYCH

Koszt komercyjnego użytkowania aplikacji nie będzie przekraczał potencjalnych zysków firmy, uzyskanych poprzez zwiększenie produktywności pracowników.

### 2.5 UNIWERSALNOŚĆ

Aplikacja dostępna jest z poziomu przeglądarki internetowej - co powoduje, że jedynym ograniczeniem w korzystaniu z niej jest posiadanie dowolnego urządzenia z dostępem do internetu.

### 2.6 ATRAKCYJNY PROJEKT WIZUALNY

Przejrzysty i elegancki interfejs zwiększa jakość produktu, korzystanie z aplikacji jest proste i intuicyjne.

### 2.7 ZAPEWNIENIE UŻYTKOWNIKOWI ROZRYWKI

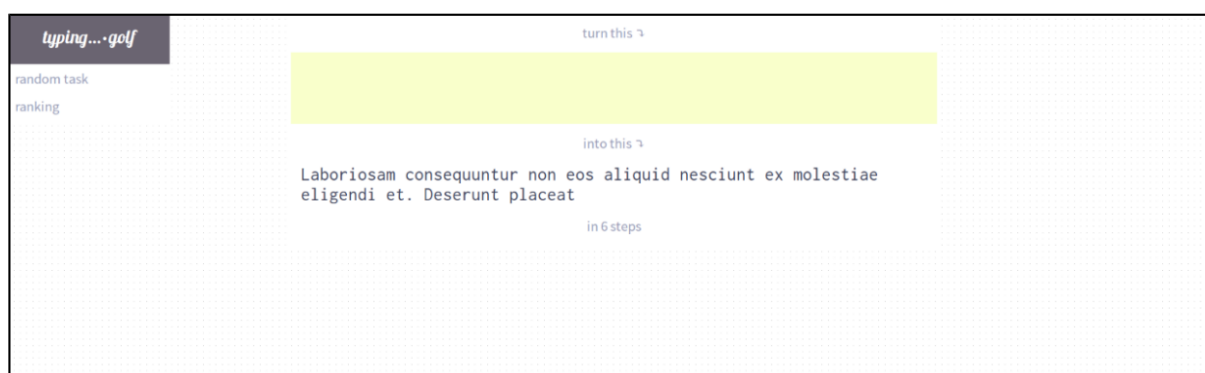
Forma, w jakiej przygotowane zostały zadania; przedstawienie punktu wyjścia oraz celu

jaki ma osiągnąć stawia przed użytkownikiem wyzwanie imitujące udział w grze.

### 3. Szczegółowy opis aplikacji

---

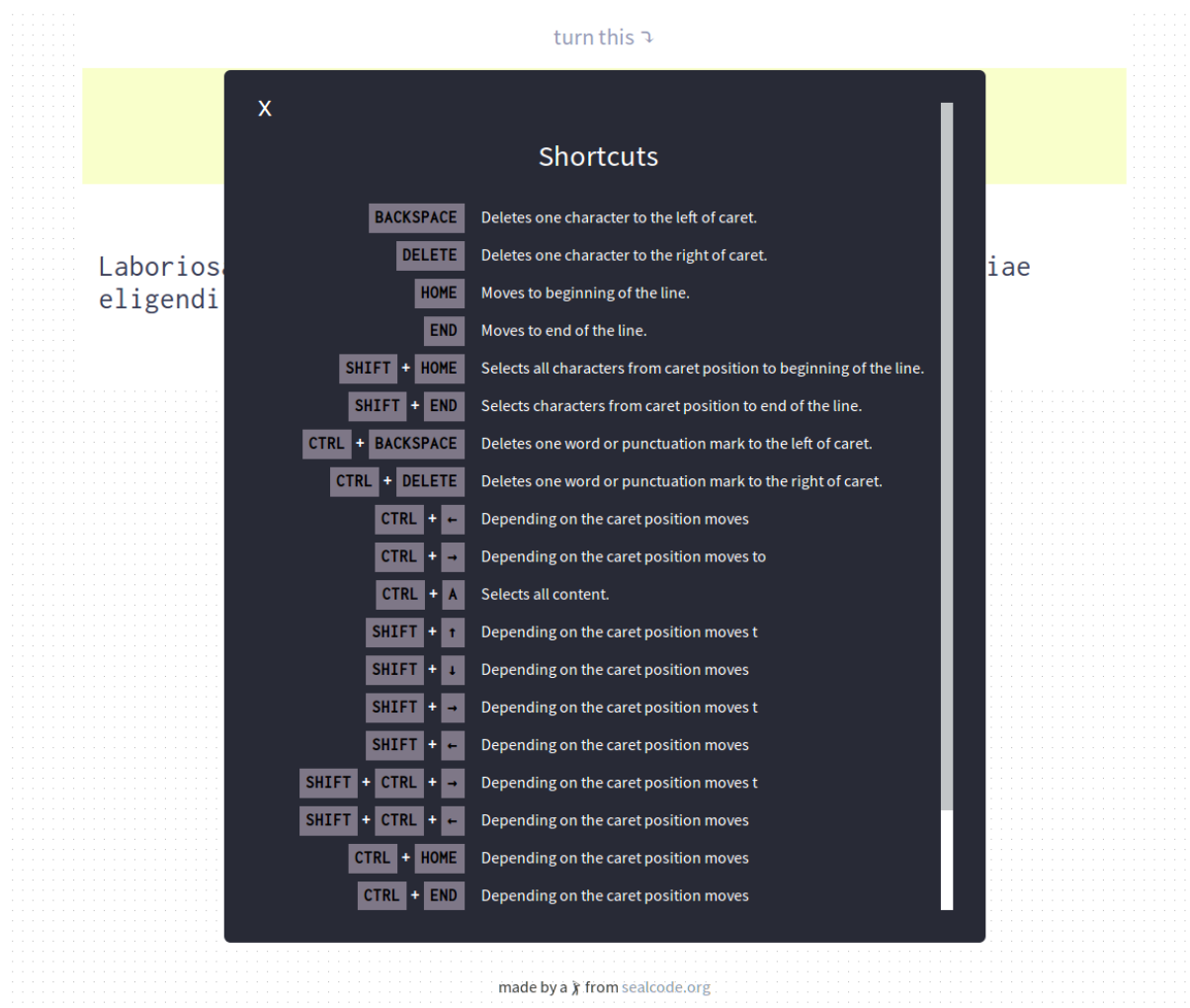
Aplikacja Typing Golf znajduje się na serwerze, dostępnym dla wszystkich internautów. Jest ona darmowa dla użytkowników prywatnych. Celem korzystania z aplikacji jest nabycie nawyku korzystania ze skrótów klawiaturowych, służących do modyfikacji i przeglądania treści tekstowych. Idea aplikacji zakłada prostą formę nauki, bazującą na czytelnym i nieskomplikowanym interfejsie.



Rys 1. Interfejs aplikacji.

Użytkownik po uruchomieniu strony internetowej z aplikacją ma bezpośredni dostęp do realizacji zadań. Każde zadanie polega na przekształceniu wygenerowanego fragmentu tekstu do podanego poniżej wzoru. Skuteczną realizację nauki korzystania ze skrótów klawiaturowych gwarantuje podana liczba kroków, w jakiej można osiągnąć daną modyfikację tekstu. Liczba kroków potrzebna do wykonania przekształcenia, obliczana jest w oparciu o algorytm bazujący na zagadnieniu teorii grafów; algorytmie przeszukiwania grafu wszerek. Testowana jest każda możliwa kombinacja ruchów, w celu wyboru opcji gwarantującej użycie jak najmniejszej liczby kroków. Aplikacja zawiera bazę zadań, wraz ze znalezionymi przez ten algorytm rozwiązaniami. Ze względu na powyższą formę realizacji wyboru zadań nie został przewidziany system rejestracji użytkowników. Częścią aplikacji

jest zakładka z opisanymi wszystkimi dozwolonymi skrótami klawiaturowymi. Użytkownik ma możliwość sprawdzenia dostępnym skrótów klawiaturowych podczas wykonywania zadania.



Rys 2. Spis skrótów klawiaturowych.

## ***4. Podział ról w zespole***

---

### **4.1 JAN ORLIK (SCRUM MASTER, DEVELOPER)**

Do odpowiedzialności Scrum Mastera należy: organizowanie spotkań, na których planuje się sprinty, retrospektywy, stand-upy itp, pośredniczenie w komunikacji pomiędzy zleceniodawcami, a programistami, usuwanie przeszkód, na jakie natrafi zespół.

### **4.2 ARKADIUSZ WIECZOREK (DEVELOPER)**

Deweloper wykonuje zadania zgodnie z taskami w Jirze. Zgłasza wszelkie napotkane problemy Scrum Masterowi. Estymuje czas wykonania tasków. Tworzy elegancki i czytelny dla reszty zespołu kod. Opisuje testerom, jak mają sprawdzić poprawność wykonanego zadania.

### **4.3 MACIEJ NOWAK (TESTER)**

Tester sprawdza tickety zdeliwerowane przez deweloperów. Sprawdza, czy dostarczony kod jest zgodny z opisem ticketa (issue). Powiadamia deweloperów, jeżeli pojawiają się jakieś niezgodności lub błędy w implementacji. Oznacza tickety jako “wykonane poprawnie” lub “do poprawki”. Opisuje klarownie, w jaki sposób wykonanie ticketa odbiega od jego opisu. W razie potrzeby załącza zrzuty ekranu lub screencasty prezentujące te rozbieżności.

### **4.4 PATRIZIA MARCINIAK (PRODUCT OWNER)**

Product Owner reprezentuje użytkowników, czyli dąży do jak najwyższej jakości końcowego produktu. Definiuje cel i sprawuje kontrolę nad rozwojem projektu, aby doszło do jego realizacji. Prezentuje obraz oczekiwanego produktu, definiuje wszystkie funkcje jakie ma spełniać. Określa, które z tych funkcji mają największy priorytet i będą najczęściej wykorzystywane oraz które funkcjonalności powinny zostać zaimplementowane jako pierwsze. Pisze historię użytkowników, czyli opis w jakich celach użytkownik będzie używał projektu, czego będzie oczekiwał, co może stanowić dla niego problem. Definiuje testy akceptacyjne. Wyjaśnia wątpliwości związane z oczekiwaniami klienta. Jest cały czas na

bieżący z rozwojem projektu. Pozostaje w kontakcie z resztą zespołu. Podejmuje optymalne decyzje biorąc pod uwagę ryzyko i ograniczenia. Jego decyzje dotyczące gotowości produktu są ostateczne.

## **5. Wymagania funkcjonalne**

---

### **5.1 ROZWIĄZYWANIE ZADANIA**

**Opis:** użytkownik wyświetla w aplikacji zadanie, wykonuje kroki mające na celu otrzymanie żądanej formy tekstu. Informacja o najmniejszej liczbie kroków potrzebnych do wykonania zadania jest dostępna na ekranie.

**Dane wejściowe:** rozwiązanie użytkownika w podanej liczbie kroków.

**Dane wyjściowe:** informacja o poprawnym rozwiązaniu zadania.

**Warunki początkowe:** połączenie z Internetem.

**Stan końcowy:** do zadania jest dopisana informacja, że zostało ono rozwiązane przez użytkownika, wraz z liczbą ruchów jakie wykonał.

### **5.2 LOSOWANIE NOWEGO ZADANIA**

**Opis:** użytkownik wybiera opcje wylosowania nowego zadania.

**Dane wejściowe:** zgłoszenie przez użytkownika chęci wykonania nowego zadania.

**Dane wyjściowe:** wyświetlenie nowego zadania.

**Warunki początkowe:** użytkownik jest podłączony do Internetu .

**Stan końcowy:** użytkownik ma możliwość wykonania nowego zadania.

### **5.3 SPRAWDZENIE DOZWOLONYCH SKRÓTÓW KŁAWIATUROWYCH**

**Opis:** użytkownik sprawdza jakie ruchy może wykonać przy pomocy skrótów klawiaturowych.

**Dane wejściowe:** zgłoszenie przez użytkownika chęci sprawdzenia listy skrótów

klawiaturowych.

**Dane wyjściowe:** wyświetlenie listy skrótów klawiaturowych.

**Warunki początkowe:** użytkownik jest podłączony do Internetu

**Stan końcowy:** wyświetlona lista skrótów

## 6. Wymagania niefunkcjonalne:

---

**Rozmiar:** serwer programu powinien uruchamiać się na niskiej klasy VPS-ie:

21GB RAM, 1 vCore (2.4GHz), 2GB RAM, 10 GB SSD

**Łatwość obsługi:** instrukcja obsługi - max 1 strona A4

**Odporność:** restart po awarii: 15 sekund. Dane bezpieczne po awarii w 100% .

(nie zostaną utracone)

**Przenośność:** dostępność w przeglądarkach: Firefox 42+, Chrome 30+, MS Edge 11.

## 7. Aktorzy systemu

---

### 7.1 AKTOR GŁÓWNY

**Twórcy** – Twórcami są członkowie zespołu

**Użytkownik** - Użytkownikiem są osoby, którym zależy na poprawie wydajności ich pracy programistycznej, pisarskiej czy edytorskiej. Cenią sobie wygodę, praktyczność i efektywność w używaniu klawiatury.

### 7.2 AKTOR POMOCNICZY

**1. System tworzenia tekstów** – Zestaw narzędzi umożliwiających wprowadzanie nowych zadań do systemu.



Aktor	Cel	Priorytet
Twórca	Zmień wygląd interfejsu	2
	Dodaj funkcjonalność	1
	Dodaj nowy skrót klawiszowy	2
Użytkownik	Losuj zadanie	1
	Rozwiąż zadanie	1
	Sprawdź dostępne skróty klawiszowe	2

Tab. 1. Zależności aktor - cel.

Akcja	In	Out
Rozwiązywanie zadania	X	
Losowanie zadania	X	
Wybór źródła tekstu zadania		X
Sprawdzanie dostępnych skrótów klawiaturowych	X	

Tab. 2. Lista In - Out

## 8. Przypadki użycia

---

### 8.1 ROZWIĄZANIE LOSOWEGO ZADANIA

**Aktor główny:** użytkownik

**Poziom:** użytkownika

**Scenariusz główny:**

1. Użytkownik uruchamia stronę internetową z aplikacją Typing Golf.
2. Użytkownik losuje zadanie.
3. Użytkownik akceptuje zadanie.
4. Użytkownik realizuje zadanie.
5. System wyświetla komunikat o poprawnym wykonaniu zadania.
6. Użytkownik otrzymuje kolejne losowo wygenerowane zadanie.

**Gwarancje minimalne:**

1. Użytkownik uruchamia stronę internetową z aplikacją.
2. Użytkownik losuje zadanie i zapoznaje się z jego treścią.

**Gwarancja powodzenia:**

1. Wyświetlony zostaje komunikat o poprawnym rozwiązaniu problemu.

**Scenariusz rozszerzony:**

1. Wystąpił błąd w połączeniu sieciowym.  
Użytkownik resetuje połączenie sieciowe.  
Użytkownik ponawia próbę logowania.
2. Sekcja wykonywania zadań niedostępna.  
Wyświetlany jest komunikat o braku dostępu do tej sekcji zadań.  
Klient akceptuje niemożność wykonania zadania.
3. Użytkownik nie chce rozwiązywać wylosowanego zadania.

Użytkownik zaznacza opcję “Losuj ponownie”.

Powrót do pkt. 3.

4. Użytkownik traci połączenie z Internetem.

Użytkownik ponownie nawiązuje połączenie i rozpoczyna realizację zadania od początku.

## **8.2 SPRAWDZENIE DOSTĘPNYCH W SKRÓTÓW KŁAWIATUROWYCH**

**Aktor główny:** użytkownik

**Poziom:** użytkownika

**Scenariusz główny:**

1. Użytkownik uruchamia stronę internetową z aplikacją Typing Golf.
2. Użytkownik wybiera zakładkę "Cheatsheet".
3. Użytkownik zostaje przekierowany do zakładki zawierającej spis dostępnych w aplikacji skrótów klawiaturowych.

**Gwarancja powodzenia:**

1. Użytkownik uzyskuje informację o dostępnych skrótach klawiaturowych

**Scenariusz rozszerzony:**

1. Wystąpił błąd w połączeniu sieciowym.  
Użytkownik resetuje połączenie sieciowe.  
Użytkownik ponawia próbę logowania.
2. Sekcja spisu dostępnych skrótów klawiaturowych niedostępna  
Wyświetlany jest komunikat o braku dostępu do tej sekcji zadań  
Klient akceptuje niemożność wykonania zadania.
3. Użytkownik traci połączenie z internetem  
Użytkownik ponownie nawiązuje połączenie i rozpoczyna realizację zadania od początku.

## 9. Plan testów

---

W celu zapewnienia jak najwyższej jakości dostarczanego produktu, aplikacja Typing Golf została przetestowana na różnych poziomach. Dobór różnorodnych testów zagwarantował wykrycie zarówno błędów w funkcjonowaniu samych metod wykorzystywanych w aplikacji, jak i błędów występujących w projekcie interfejsu.

### 9.1. ZAKRES TESTÓW

#### 9.1.1. WŁAŚCIWOŚCI TESTOWANE

- uruchamianie strony internetowej, na której znajduje się aplikacja
- dostęp do wszystkich komponentów aplikacji
- możliwość wykonania zadania
- skutek wykonania zadania
- przejrzystość i zrozumiałość aplikacji
- merytoryczna poprawność kodu

#### 9.1.2. WŁAŚCIWOŚCI NIETESTOWANE

- odporność na wystąpienie awarii serwera
- czas odpowiedzi serwera
- autentyczność minimalnej liczby kroków potrzebnej do wykonania zadania

## 9.2. STRATEGIA TESTOWANIA - TYPY TESTÓW

### **TESTY JEDNOSTKOWE FUNKCJONALNOŚCI**

**Cel:**

Sprawdzenie czy poszczególne komponenty kodu aplikacji funkcjonują w sposób zgodny z oczekiwanym.

**Definicja błędu:**

Test jednostkowy nie zwraca oczekiwanego wyniku.

**Skala ocen:**

Test jednostkowy może zwrócić jako wynik informację o poprawnym wykonaniu testu lub wystąpieniu błędu podczas jego wykonywania.

**Kryterium pozytywnego zakończenia testu:**

W przypadku testów jednostkowych dostarczona zostaje informacja, czy dany fragment kodu wykonuje swoją funkcjonalność, w sposób oczekiwany przez twórców. Za test pozytywnie ukończony zostaje uznany test, zwracający informację o poprawnym wykonaniu kodu.

### **Testy użyteczności**

**Cel:**

Uzyskanie informacji, czy aplikacja, jest zrozumiała i atrakcyjna dla użytkownika.

**Definicja błędu:**

Tester negatywnie ocenia jedną z testowanych cech lub funkcjonalności aplikacji

**Skala ocen:**

Tester ocenia w skali od 0 do 5, gdzie 0 jest notą najniższą, natomiast 5 najwyższą:

- zrozumiałość funkcji aplikacji
- dostęp do komponentów

- responsywność aplikacji

**Kryterium pozytywnego zakończenia testu:**

Średnia ocena wszystkich testowanych aspektów działania aplikacji oscyluje w przedziale 3-5.

## Testy konfiguracji

**Cel:**

**Część 1.** Uzyskanie informacji czy aplikacja uruchamia się w różnych przeglądarkach internetowych .

**Część 2.** Ocena prezencji komponentów wizualnych przy uruchamianiu aplikacji na urządzeniach z ekranami o różnej rozdzielczości.

**Definicja błędu:**

**Część 1.** Aplikacja nie uruchamia się lub uruchamia się w sposób uniemożliwiający jej użytkowanie.

**Część 2.** Komponenty wizualne aplikacji rozmieszczone są w sposób niezgodny z określonym lub w sposób uniemożliwiający użytkowanie lub wydajne użytkowanie aplikacji.

**Skala ocen:**

**Część 1.** Ocena pozytywna przy prawidłowym uruchomieniu aplikacji, ocena negatywna przy braku możliwości skorzystania z aplikacji

**Część 2.** Ocena w skali od 0 do 5, gdzie 0 to sposób umieszczenia komponentów w sposób uniemożliwiający korzystanie z aplikacji, a 5 to widok zgodny z określonym w specyfikacji.

**Kryterium pozytywnego zakończenia testu:**

**Część 1.** Aplikacja uruchamia się poprawnie

**Część 2.** Uzyskana nota w przedziale 3-5.

## Testy systemowe

### **Cel:**

Sprawdzenie czy następuje uruchomienie aplikacji.

Sprawdzenie czy wystąpiły błędy podczas uruchamiania kodu aplikacji.

### **Definicja błędu:**

System nie uruchamia się.

W konsoli przeglądarki po uruchomieniu aplikacji wystąpiły błędy.

### **Skala ocen:**

Ocena negatywna jeśli system nie uruchomił się lub uruchomił się, ale wystąpiły błędy uniemożliwiające jego użytkowanie.

Ocena pozytywna w przypadku uruchomienia aplikacji bez wystąpienia błędów.

Ocena pozytywna z uwagami jeśli system uruchomił się i wystąpiły błędy nie uniemożliwiające korzystania z aplikacji.

### **Kryterium pozytywnego zakończenia testu:**

Uzyskanie oceny pozytywnej.

## 9.3. ZASOBY TESTOWE

### Testy jednostkowe

**Wykonawca:** programista – członek zespołu

### **Zasoby:**

- środowisko programistyczne JavaScript
- kod aplikacji
- pakiety programistyczne służące do testowania jednostkowego (JS assert)

- ponad 200 testowanych przypadków

## Testy użyteczności

**Wykonawca:** potencjalny użytkownik – student kierunku informatyka

### **Zasoby:**

- komputer z dostępem do Internetu
- test ewaluacyjny

## Testy konfiguracji

**Wykonawca:** programista – członek zespołu

### **Zasoby:**

- przeglądarki internetowe Firefox 42+, Chrome 30+, MS Edge 11
- urządzenia o następujących rozdzielczościach ekranu: 1366 × 768, 3200 × 1800, 1920×1080

## Testy systemowe

**Wykonawca:** programista – członek zespołu

### **Zasoby:**

- przeglądarka testująca aplikacje internetowe PhantomJS
- dostęp do Internetu



## 9.4. SPECYFIKACJA TESTÓW

### 9.4.1 SCENARIUSZE PRZYPADKÓW TESTOWYCH

#### Testy jednostkowe

1. Uruchomienie testu odpowiadającego funkcjonowaniu pojedynczego komponentu kodu.
2. Określenie wyniku testu.

#### Testy użytkownika

1. Przedstawienie testerowi funkcji aplikacji.
2. Przedstawienie dostępnych komponentów.
3. Określenie wymagań – rozwiązania jednego z oferowanych przez aplikację zadań.
4. Zapoznanie się dostępnymi elementami aplikacji
5. Wystawienie przez testera opinii.

#### Testy konfiguracji

##### **Część 1.**

1. Uruchomienie aplikacji w przeglądarkach Firefox 42+, Chrome 30+, MS Edge 11
2. Ocena czy strona otwiera się prawidłowo.
3. Ocena czy elementy zostają uruchomione w sposób umożliwiający korzystanie z aplikacji.

##### **Część 2.**

1. Uruchomienie aplikacji na urządzenia o rozdzielczościach ekranu: 1366 × 768, 3200 ×

1800, 1920×1080.

2. Ocena czy rozmieszczenie elementów umożliwia korzystanie z aplikacji.
3. Ocena czy rozmieszczenie elementów jest czytelne.
4. Ocena czy rozmiar czcionki i komponentów jest dostosowany do rozdzielczości ekranu.
5. Wystawienie oceny na podstawie powyższych kryteriów.

## Testy systemowe

1. Uruchomienie skryptu PhantomJS.
2. Testowane parametry: poprawne uruchomienie strony.
3. Analiza wyprowadzonego wyniku.

### 9.4.2. PRZYKŁADOWE RAPORTY TESTOWANIA:

## Testy jednostkowe

Działanie	Wynik
Uruchomienie testu	-
Wyprowadzony wynik	pozytywny

## Testy użytkownika

Działanie	Wynik
Przedstawienie testerowi zasad działania aplikacji	Tester rozumie funkcje aplikacji

Tester wykonuje zadanie i zapoznaje się z komponentami aplikacji	4
Tester dokonuje ewaluacji zrozumiałości interfejsu aplikacji	4
Tester dokonuje ewaluacji dostęp do komponentów	5
Tester dokonuje ewaluacji responsywności aplikacji	3
Zakończenie testu	4 - Średnia ocena wszystkich testowanych aspektów funkcjonalności i interfejsu

## Testy konfiguracji

Działanie	Wynik
Uruchomienie aplikacji w przeglądarce Firefox 42+	Uruchomienie aplikacji powiodło się
Uruchomienie aplikacji w przeglądarce Chrome 30+	Uruchomienie aplikacji powiodło się
Uruchomienie aplikacji w przeglądarce MS Edge 11	Uruchomienie aplikacji powiodło się

Działanie	Wynik
Uruchomienie aplikacji na urządzeniu o rozdzielczości ekranu 1366 × 768	Uruchomienie aplikacji powiodło się
Ocena czytelności rozmieszczenia komponentów aplikacji	4

Ocena funkcjonalności rozmieszczenia komponentów aplikacji	5
Ocena dostosowania rozmiaru czcionki i komponentów do rozmiaru okna	4
	Ocena ogólna: 4.34
Uruchomienie aplikacji na urządzeniu o rozdzielczości ekranu $1366 \times 768$ , $3200 \times 1800$ , $1920 \times 1080$ .	Uruchomienie aplikacji powiodło się
Ocena rozmieszczenia komponentów aplikacji	5
Ocena funkcjonalności rozmieszczenia komponentów aplikacji	5
Ocena dostosowania rozmiaru czcionki i komponentów do rozmiaru okna	5
	Ocena ogólna: 5
Uruchomienie aplikacji na urządzeniu o rozdzielczości ekranu $1366 \times 768$ , $3200 \times 1800$ , $1920 \times 1080$ .	Uruchomienie aplikacji powiodło się
Ocena rozmieszczenia komponentów aplikacji	5
Ocena funkcjonalności rozmieszczenia komponentów aplikacji	5
Ocena dostosowania rozmiaru czcionki i komponentów do rozmiaru okna	4
	Ocena ogólna: 4.67

## Testy systemowe

Działanie	Wynik
Uruchomienie skryptu PhantomJS	-
Wynik	System poprawnie się uruchomił Brak błędów

## 10. Ocena jakości systemu

Jakość systemu została oceniona na podstawie schematu CUPRIMDA, przez zespół składający się z twórców aplikacji. Dołożyliśmy starań, aby wystawianie noty nie były motywowane faktem oceniania własnego produktu, a obiektywną oceną systemu. W celu ewaluacji jakości systemu każdy członek określił w skali od 0 do 10 (gdzie nota 0 oznacza ocenę najniższą, 10 najwyższą) w jakim stopniu aplikacja realizuje cechy wchodzące w skład schematu CUPRIMDA.

CUPRIMDA	
<b>CAPABILITY</b>	funkcjonalność
<b>USABILITY</b>	łatwość obsługi
<b>PERFORMANCE</b>	wydajność
<b>RELIABILITY</b>	niezawodność
<b>INSTALLABILITY</b>	łatwość instalacji
<b>MAINTAINABILITY</b>	łatwość utrzymania
<b>DOCUMENTATION</b>	dokumentacja
<b>AVAILABILITY</b>	dostępność

Tab. 3. Elementy schematu CUPRIMDA

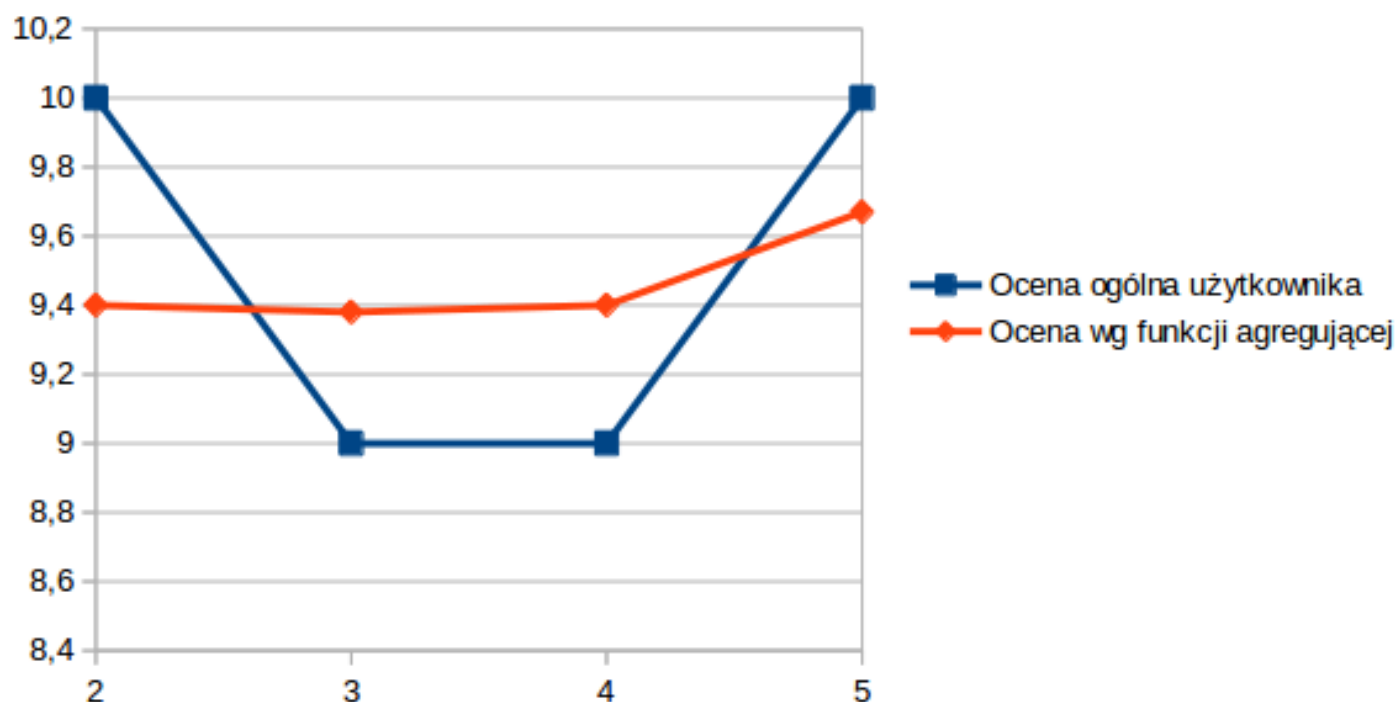
	Jan Orlik	Patrizia Marciniak	Maciej Nowak	Arkadiusz Wieczorek	Średnia ocena
<b>Capability</b>	10	8	9	10	<b>9,25</b>
<b>Usability</b>	9	10	10	9	<b>9,5</b>
<b>Performance</b>	8	7	8	9	<b>8</b>
<b>Reliability</b>	10	10	10	10	<b>10</b>
<b>Installability</b>	10	10	10	10	<b>10</b>
<b>Maintainability</b>	9	10	10	10	<b>9,75</b>
<b>Documentation</b>	7	8	7	9	<b>7,75</b>
<b>Availability</b>	10	9	10	9	<b>9,5</b>
<b>Ocena ogólna użytkownika</b>	<b>10</b>	<b>9</b>	<b>9</b>	<b>10</b>	<b>9,5</b>
Odchylenie od średniej	0,5	0,5	0,5	0,5	
<b>Ocena wg funkcji agregującej</b>	<b>9,4</b>	<b>9,4</b>	<b>9,58</b>	<b>9,67</b>	<b>9,5125</b>
Odchylenie od średniej	0,1125	0,1125	0,0675	0,1575	

Tab. 4. Ocena użytkowników

	<b>Wagi</b>
<b>Capability</b>	9
<b>Usability</b>	10
<b>Performance</b>	6
<b>Reliability</b>	18
<b>Installability</b>	23
<b>Maintainability</b>	17
<b>Documentation</b>	7
<b>Availability</b>	10
<b>Suma</b>	<b>100</b>
<b>Współczynnik korelacji:</b>	0,96225

Tab. 5. Funkcja agregująca: średnia ważona. Wagi przypisane poszczególnym elementom.

### Korelacja ocen użytkowników i funkcji agregującej



## 11. Wymagania sprzętowe

---

Dostęp do aplikacji jest możliwy poprzez dowolne urządzenie posiadające klawiaturę oraz dostęp do Internetu.

## 12. Plan badania użyteczności

---

### 12.1. WPROWADZENIE

Plan badania użyteczności aplikacji Typing Golf został sporządzony przez twórców aplikacji. Miał on na celu przetestowanie poziomu użyteczności naszej aplikacji dla różnych grup odbiorców.

### 12.2 UCZESTNICY

- a. Marcin Orlik - pisarz fantastyki - jako twórca prozy z chęcią poznaje nowe sposoby, które pozwalają uczynić proces pisanie jak najbardziej bezbolesnym

b. 5 członków Fundacji Kultury Akademickiej - jest to fundacja wspierająca środowisko pisarzy w Poznaniu. Mieszczą się w grupie docelowej Typing Golf-a, ponieważ jako członkowie fundacji muszą tworzyć duże ilości dokumentów tekstowych. Wezmą oni udział w badaniu rozszerzonym

## 12.3 METODOLOGIA

Badanie użyteczności polega na bezpośrednim użyciu gotowej aplikacji przez testową grupę użytkowników. Badanie podzielone jest na dwie części, pierwsza opiera się na przeprowadzeniu lekcji produktywnego używania klawiatury poprzez wykonania zadania pochodzącego z aplikacji dla licznej (30 osób) grupy ankietowanych. Następnie są oni poproszeni o wypełnienie ankiety. Druga grupa testowanych osób (5 osób), wzięła udział w rozszerzonym badaniu, które polegało na wykonywaniu jednego zadania z aplikacji dziennie przez okres tygodnia. Druga część ma na celu sprawdzenie, jak skuteczne jest nabywanie nawyków poprawnego pisania podczas użytkowania naszej aplikacji ze średnią częstotliwością. W celu obiektywnej oceny, ankietowani zostali poddani dwóm testom (na początku oraz końcu okresu testowania) polegającym na przekształceniu podanego tekstu do zadanej postaci. Oba zadania wymagały zastosowania takich samych metod edycji tekstu, w celu uzyskania wiarygodnych wyników badań. Postęp użytkownika oceniony został na podstawie tego, ile i jakie ruchy wykonał w celu realizacji zadania.

## 12.4 CEL BADANIA

Wyniki badania mają umożliwić poprawienie funkcjonowania aplikacji w dwóch kontekstach. Pierwszy główny cel to zwiększenie zysków dla użytkownika poprzez wprowadzenie zmian w aplikacji, które umożliwią szybsze wyrabianie dobrych nawyków szybkiego używania skrótów klawiszowych. Dzięki uwagom i przeprowadzonym testom będziemy mogli wdrożyć rozwiązania, usprawniające działanie aplikacji. Dzięki grupie testerów korzystając z aplikacji przez okres dwóch tygodni uzyskaliśmy informację, które skróty klawiszowe przyswoili najlepiej, a które powinny zostać wprowadzone do ćwiczeń z większą częstotliwością. Dodatkowym zyskiem może okazać się wdrożenie zmian dotyczących interfejsu aplikacji, czy sposobu interakcji z użytkownikiem w oparciu o uwagi użytkowników opisane w ankiecie.



## 12.5 SCENARIUSZ BADANIA:

### a. wykonanie zadania + ankieta

- Omówienie zadania: wyjaśnienie jak działa aplikacja, jak rozpocząć wykonywanie zadania, jaki jest cel wykonania zadania
- Wykonanie zadanie
- Omówienie ankiety: ankieta nie jest anonimowa, dzięki uzyskanym informacjom jakie uwagi pochodziły od jakiej grupy osób, dowiadujemy się, poprawienie których aspektów działania aplikacji pozwoli zwiększyć grono odbiorców; na każde pytanie zaznaczamy odpowiedz z przedziału 1-5 gdzie liczby oznaczają:

1 zdecydowanie nie

2 raczej nie

3 nie mam zdania

4 raczej tak

5 zdecydowanie tak

- Przedstawienie regulaminu uczestnictwa w ankiecie
- Przeprowadzenie ankiety

### b. test oparty o codzienną interakcje z systemem

Część 1. Spotkanie z testowanymi:

- wyjaśnienie na czym polega test
- omówienie zadania mającego na celu sprawdzić biegłość badanych w użytkowaniu klawiatury
- omówienie działania aplikacji
- omówienie regulaminu udziału w badaniu

Część 2. Codzienne wykonywanie zadania przez okres tygodnia.

Część 3. Spotkanie z testowanymi:

- ponowne wykonanie zadania sprawdzającego
- przeprowadzenie ankiety (ta sama co pierwsza grupa)
- zebranie opinii na temat produktu

**Ankieta: (ocenić w skali od 1 do 5, gdzie 1 to nota najniższa a 5 najwyższa)**

1. Czy interfejs aplikacji jest intuicyjny?
2. Czy Interfejs aplikacji podoba Ci się wizualnie?
3. Czy obsługa aplikacji sprawiła Ci problem?
4. Czy uważasz, że przykłady z zadania z którym musiałeś się zmierzyć były trudne?
5. Czy uważasz, że przykłady z zadania są przydatne w codziennym używaniu klawiatury?
6. Czy zapamiętałeś jakieś skróty klawiszowe, których użyłeś podczas wykonywania zadania?
7. Czy chciałbyś wykonać inne zadania z aplikacji Typing Golf?
8. Czy poleciłbyś aplikację Typing Golf znajomym?
9. Czy jest coś co zmieniłbyś w aplikacji?
10. Jeśli na powyższe pytanie zaznaczyłeś 4 lub 5, napisz swoje uwagi.

**Zadania tekstowe:**

Przekształć tekst tylko przy użyciu klawiatury za pomocą jak najmniejszej liczby kroków

Litwo! Ojczyzna moja! ty jesteś jak zdrowie:

Ile cie trzba cnieć, ten tylko się dowie,

Kto cię stracił. Dziś piękność twą w całej ozdobie widze i opisuje bo tęsknię po tobie.

Do postaci:

Litwo! Ojczyzno moja! ty jesteś jak zdrowie:  
Ile cię trzeba cenić, ten tylko się dowie,  
Kto cię stracił. Dziś piękność twą w całej ozdobie  
Widzę i opisuję, bo tęsknię po tobie.

### ***13. Instrukcja obsługi dla użytkownika:***

---

Losuj zadanie

Przejdź do rankingu

Tekst startowy

*typing...•golf*

random task

ranking

Ala ma kota

you've done 5 steps

into this ↴

Ala ma psa

in 6 steps

made by a ↵ from sealcode.org



Tekst docelowy

Wyświetl listę skrótów

Licznik wykonanych kroków

Minimalna ilość kroków do wykonania zadania