



POLSKO-JAPOŃSKA
AKADEMIA TECHNIK
KOMPUTEROWYCH

Kierunek studiów: Informatyka

Rodzaj studiów: Zaoczne

Praca dyplomowa

Temat pracy: Fishki - multiplatformowa aplikacja służąca do nauki pamięciowej

Temat w języku angielskim: Fishki - a multiplatform application for memory learning

Opiekun pracy: dr Puźniakowski Tadeusz

Wykonawcy:

Nazwisko, imie	Nr albumu
Kossak Oliwier	s22018
Klimowski Daniel	s18504
Krieger Wiktor	s23638
Żurawski Jakub	s23047

Streszczenie:

DO ZROBIENIA NA KOŃCU - NA RÓWNI ZE WSTĘPEM

Gdańsk, 20 marca 2024



POLSKO-JAPOŃSKA
AKADEMIA TECHNIK
KOMPUTEROWYCH

Karta projektu

Temat projektu: Fishki - multiplatformowa aplikacja służąca do nauki pamięciowej		Akronim: Fishki Data ustalenia tematu 11.10.2023
Promotor: dr Puźniakowski Tadeusz		Konsultanci: 1. — brak —
Cele projektu: Wytwarzany system przynosi rozwiązanie służące do efektywnej nauki pamięciowej. Cel projektu odpowiada na problem rozumiany jako trudności w organizacji oraz korzystania z materiałów służących do opanowywania pojęć.		
Rezultaty i zakres	Oczekiwane produkty/usługi: Strona internetowa, aplikacja mobilna, serwer obsługujący utrzymanie i żywotność produktu Główne funkcjonalności i/lub cechy: Nauka metodą fiszek, obsługa talii tj. zestawów fiszek, obsługa talii fiszek za pomocą mowy, tworzenie fiszek z pomocą sztucznej inteligencji.	
Miary sukcesu: Wytworzenie działającej strony internetowej i aplikacji mobilnej, opracowanie techniczne projektu z wykorzystaniem infrastruktury serwerowej, zaimplementowanie silnika sztucznej inteligencji, zrealizowanie wymagań systemowych na poziomie ‘wymagane’.		
Ograniczenia: Deadline, zdalny charakter pracy nad projektem, budżet, brak doświadczenia w pracy nad projektem o zadanej złożoności, nauka i wykorzystanie nowych technologii.		

Wykonawcy	Numer albu- mu	Specjalizacja	Tryb studiów
Kossak Oliwier	s22018	Sztuczna Inteligencja	Niestacjonarny
Klimowski Daniel	s18504	Sztuczna Inteligencja	Niestacjonarny
Krieger Wiktor	s23638	Sztuczna Inteligencja	Niestacjonarny
Żurawski Jakub	s23047	Sztuczna Inteligencja	Niestacjonarny

Oświadczenie autorów pracy dyplomowej

Świadomi odpowiedzialności prawnej oświadczamy, że niniejszą pracę dyplomową w zakresie przedstawionym przez nasz zespół projektowy wykonaliśmy samodzielnie i nie zawiera ona treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczamy również, że praca w przedstawionym przez nas zakresie nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu ukończenia studiów wyższych.

Oświadczamy ponadto, że niniejsza wersja pracy dyplomowej jest identyczna z załączoną wersją elektroniczną.

Spis treści

1	Informacje wstępne	5
1.1	O projekcie	5
2	Przykładowe elementy	6
2.1	Listingi	6
2.2	Obrazki	6
3	Karty udziałowca	8
4	Wymagania wszelakie	9

Rozdział 1

Informacje wstępne

1.1 O projekcie

DO ZROBIENIA NA KOŃCU

Rozdział 2

Przykładowe elementy

W niniejszym rozdziale pokażę różne ciekawe elementy które można użyć w pracy. Nie są to wszystkie możliwości, ale te, które moim zdaniem najczęściej pojawiają się w pracach inż. Na przykład, jeśli chcemy powołać się na jakieś źródła, robimy to tak [BEHESHTIROUI2021107419].

2.1 Listingi

Jak załączać kod źródłowy jest pokazane na listingu 2.1

```
1 printf("hello");
```

Listing 2.1: Przykładowy witaj w świecie

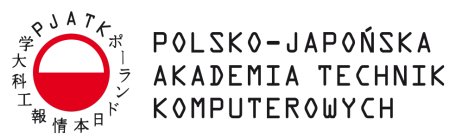
2.2 Obrazki

Na ilustracji 2.1 widzimy oficjalne logo PJATK.



Rysunek 2.1: Logo PJATK załączone jako obrazek

albo dla wygody jako makro tak jak na obrazku 2.2



Rysunek 2.2: Obrazek załączony za pomocą makra

Rozdział 3

Karty udziałowca

KARTA UDZIAŁOWCA	
Identyfikator:	jednoznaczny symbol np. UOB 01, UOB 02 ... dla udziałowców ożywionych bezpośrednich, UNP 01... dla nieożywionych pośrednich itd.
Nazwa::	nazwa udziałowca
Opis::	opis udziałowca
Typ udziałowca::	ożywiony/nieożywiony, bezpośredni/pośredni
Punkt widzenia::	z jakiej perspektywy patrzy udziałowiec np. technicznej, ekonomicznej, operatora systemu itp.
Ograniczenia::	ograniczenia udziałowca np. administrator nie powinien specyfikować wymagań finansowych
Wymagania::	tu tylko symbole wymagań wyspecyfikowanych w rozdziale 3

Tabela 3.1: Przykładowy opis udziałowca

Rozdział 4

Wymagania wszelakie

Na tabeli 4.1 pokazano jak można definiować wymagania ogólne lub dziedzinowe.

Teraz czas na wymagania funkcjonalne, na przykład 4.2

Natomiast tabela 4.3 pokazuje wymagania na interfejs z otoczeniem.

Tabela dotycząca wymagań pozafunkcyjnych 4.4 jest także widoczna.

KARTA WYMAGANIA			
Identyfikator:	jednoznaczny symbol np. WO1, WO2 ..	Priorytet:	ważność wymagań, np. wg skali MoSCoW: M – must (musi być) S – should (powinno być) C – could (może być) W – won't (nie będzie – nie będzie implementowane w danym wydaniu, ale może być rozpatrzone w przyszłości)
Nazwa:	krótki opis		
Opis:	opis szczegółowy, należy dążyć do tego, żeby wszystkie znane na ten moment szczegóły wymagania zostały wydobyte i wyspecyfikowane		
Udziałowiec:	nazwa udziałowca, który podał wymaganie		
Wymagania powiązane:	wymagania zależne i uszczegóławiające – odesłanie poprzez identyfikator		

Tabela 4.1: Przykładowe wymaganie ogólne lub dziedzinowe

KARTA WYMAGANIA			
Identyfikator:	jednoznaczny symbol np. FO1, FO2 ..	Priorytet:	ważność wymagań, np. wg skali MoSCoW: M – must (musi być) S – should (powinno być) C – could (może być) W – won't (nie będzie – nie będzie implementowane w danym wydaniu, ale może być rozpatrzone w przyszłości)
Nazwa:	krótki opis		
Opis:	<p>opis szczegółowy, należy dążyć do tego, żeby wszystkie znane na ten moment szczegóły wymagania zostały wydobyte i wyspecyfikowane Można zastosować opis jak w User Story</p> <ul style="list-style-type: none"> • Jako.. (konkretny użytkownik systemu) • chcę... (pożądana cecha lub problem, który trzeba rozwiązać) • bo wtedy/ponieważ... (korzyść płynąca z ukończenia story) 		
Kryteria akceptacji:	Warunki Satysfakcji (Szczegóły dodane na potrzeby testów akceptacyjnych)		
Dane wejściowe:	uzupełniane w trakcie sprintu – dane wejściowe, związane z wymaganiem		
Warunki początkowe:	uzupełniane w trakcie sprintu – warunki, które muszą być prawdziwe przed wywołaniem operacji		
Warunki końcowe:	uzupełniane w trakcie sprintu – warunki, które muszą być prawdziwe po wywołaniu operacji		
Sytuacje wyjątkowe:	uzupełniane w trakcie sprintu – niepożądane sytuacje i sposoby ich obsługi		
Szczegóły implementacji:	uzupełniane w trakcie sprintu – opis sposobu realizacji		
Udziałowiec:	nazwa udziałowca, który podał wymaganie		
Wymagania powiązane:	wymagania zależne i uszczegóławiające – odesłanie poprzez identyfikator		

Tabela 4.2: Przykładowa tabela z wymaganiami na interfejs z otoczeniem

KARTA WYMAGANIA			
Identyfikator:	jednoznaczny symbol np. IO1, IO2 ..	Priorytet:	ważność wymaga- nia, np. wg skali Mo- SCoW: M – must (musi być) S – should (powinno być) C – could (może być) W – won't (nie bę- dzie – nie będzie imple- mento- wane w danym wydaniu, ale może być roz- patrzone w przy- szłości)
Nazwa:	krótki opis		
Opis:	opis szczegółowy, należy dążyć do tego, żeby wszystkie znane na ten moment szczegóły wymagania zostały wy- dobyte i wyspecyfikowane		
Kryteria akceptacji:	Warunki Satysfakcji (Szczegóły dodane na potrzeby te- stów akceptacyjnych)		
Dane wejściowe:	uzupełniane w trakcie sprintu – dane wejściowe, związane z wymaganiem		
Warunki początkowe:	uzupełniane w trakcie sprintu – warunki, które muszą być prawdziwe przed wywołaniem operacji		
Warunki końcowe:	uzupełniane w trakcie sprintu – warunki, które muszą być prawdziwe po wywołaniu operacji		
Sytuacje wyjątkowe:	uzupełniane w trakcie sprintu – niepożądane sytuacje i sposoby ich obsługi		
Szczegóły implementacji:	uzupełniane w trakcie sprintu – opis sposobu realizacji		
Udziałowiec:	nazwa udziałowca, który podał wymaganie		
Wymagania powiązane:	wymagania zależne i uszczegóławiające – odesłanie po- przez identyfikator		

Tabela 4.3: Przykładowa tabela z wymaganiami na interfejs z otoczeniem

KARTA WYMAGANIA			
Identyfikator:	jednoznaczny symbol np. NFO1, NFO2 ..	Priorytet:	ważność wymagań, np. wg skali MoSCoW: M – must (musi być) S – should (powinno być) C – could (może być) W – won't (nie będzie – nie będzie implementowane w danym wydaniu, ale może być rozpatrzone w przyszłości)
Nazwa:	krótki opis		
Opis:	opis szczegółowy, należy dążyć do tego, żeby wszystkie znane na ten moment szczegóły wymagania zostały wydobyte i wyspecyfikowane		
Kryteria akceptacji:	Warunki Satysfakcji (Szczegóły dodane na potrzeby testów akceptacyjnych)		
Udziałowiec:	nazwa udziałowca, który podał wymaganie		
Wymagania powiązane:	wymagania zależne i uszczegóławiające – odesłanie poprzez identyfikator		

Tabela 4.4: Przykładowa tabela z wymaganiami pozafunkcyjnymi