

Organizacja i infrastruktura projektu

Mateusz Fydrych 193410, Damian Trowski 193443, Maciej Raciniewski 189774

I Opis projektu i produktu

Nazwa projektu:

Opracowanie podsystemów sprawdzających zadania dla różnych języków dla Systemu STOS.

Adresowany problem:

Konieczność efektywnego sprawdzania zadań programistycznych w różnych językach w kontekście dydaktycznym.

Obszar zastosowania:

Edukacja akademicka, w szczególności na wydziale WETI.

Rynek:

Akademickie środowisko edukacyjne.

Interesariusze:

Kadra dydaktyczna, studenci, opiekun projektu, zespół deweloperski, uczestnicy konkursów.

Użytkownicy i ich potrzeby:

Kadra wymagająca narzędzi do oceny zadań, studenci oczekujący sprawiedliwego i szybkiego systemu oceniania.

Cel i zakres produktu:

Celem projektu jest Opracowanie nowych podsystemów sprawdzających zadania w różnych językach programowania dla systemu stos wykorzystywanego w procesie dydaktycznym na naszym wydziale. Poszczególne podsystemy powinny zostać zrealizowane w formie oddzielnych kontenerów tak by wykorzystać modularność

głównego systemu a z drugiej strony by umożliwić efektywne skalowanie całego rozwiązania. W ramach pracy możliwe będzie również rozwinięcie obecnych rozwiązań pozwalających na zrównoleglanie sprawdzania zadań zgłoszonych do systemu.

Ograniczenia:

Zgodność z istniejącą infrastrukturą systemu STOS, czas realizacji do grudnia 2025.

Inne współpracujące systemy:

Docker, GitHub, DockerHub.

Termin:

Projekt ma zostać zakończony do grudnia 2025.

Główne etapy projektu:

Analiza, planowanie, projektowanie, implementacja, testowanie, integracja, dokumentacja, walidacja benefitów, wdrożenie

II Interesariusze i użytkownicy

Interesariusze:

Studenci, kadra dydaktyczna, zespół deweloperski, opiekun projektu dr inż. Krzysztof Manuszewski.

Użytkownicy końcowi:

Kadra akademicka korzystająca z systemu do oceny, studenci korzystający z systemu do sprawdzania zadań.

Krótki opis i klasyfikacja interesariuszy:

Interesariusze są bezpośrednio związani z procesem dydaktycznym i wynikami projektu, ich zadowolenie jest kluczowe dla sukcesu.

III Zespół

Skład zespołu i odpowiedzialność:

Mateusz Fydrych (Project Manager): zarządza projektem, dokumentacją i jest pierwszą osobą kontaktu.

Damian Trowski (Software Developer): odpowiada za rozwój oprogramowania, skuteczne rozwiązania napotkanych problemów.

Maciej Raciniewski (DevOps Engineer): zarządza infrastrukturą, operacjami i proponuje kreatywne nowe pomysły.

Tryb pracy:

Projekt realizowany w modelu hybrydowym.

Dane kontaktowe osób w zespole:

Dostępne w systemie komunikacji wewnętrznej.

Kontakt z kierownikiem:

S193410@student.pg.eti.edu.pl

IV Komunikacja w zespole i z interesariuszami

Organizacja spotkań:

Regularne spotkania wraz z osiągniętymi nowymi rezultatami.

Docelowa organizacja spotkań:

Tygodniowe spotkania sprintów.

Środki i sposoby komunikacji:

Discord, e-mail, Messenger, GitHub issues, Microsoft Teams, werbalna, niewerbalna.

Komunikacja zdalna:

Głównie przez Discorda i Messengera.

Komunikacja z otoczeniem projektu:

Regularne spotkania z opiekunem projektu przez Microsoft Teams.

V Współdzielenie dokumentów i kodu

Sposób wymiany dokumentów i kodu: GitHub, Discord.

Adres i sposób dostępu do repozytorium:

Repozytoria na GitHubie:

<https://github.com/d4m14n-gh/ProjektGrupowy>

https://github.com/d4m14n-gh/Projekt_Inzynierski-2025

Osoba odpowiedzialna za utworzenie i przygotowanie repozytorium:

Damian Trowski.

Osoba odpowiedzialna za konfigurację i utrzymanie repozytorium:

Mateusz Fydrych, Maciej Raciniewski.

Osoba odpowiedzialna za porządek w dokumentacji:

Mateusz Fydrych.

Schemat nazewnictwa dokumentów/plików:

projektGrupowy[NazwaDokumentu].pdf

Sposób wersjonowania dokumentacji:

ręcznie przez zmianę nazwy.

VI Narzędzia

Narzędzia wspierające obszary z punktów 4 i 5:

Discord, Microsoft Teams, Messenger, GitHub, Docker, DockerHub.

Narzędzia wspomagające organizację projektu i zarządzanie nim:

Github, Discord. Narzędzia do tworzenia dokumentów: Microsoft Office.

Narzędzia do modelowania:

Paint, ExceliDraw, Enterprise Architect Lite.

Narzędzia do wytwarzania i testowania systemu:

Docker, Visual Studio Code.