SRiR/2017-18/Projekt

SYSTEMY RÓWNOLEGŁE I ROZPROSZONE

PROJEKT

2018-2019 semestr zimowy

# Wstęp

## 1.1 Python



» Przejrzysta i zwięzła składnia – dobra czytelność i klarowność kodu   
» Rozbudowany pakiet bibliotek standardowych  
» Wspierane paradygmaty: obiektowy, imperatywny, funkcyjny   
» Dynamiczny system typów oraz automatyczna zarządzanie pamięcią   
» Język interpretowany – wymaga instalacji interpretera

# 2.Wykorzystane narzędzia

## 2.1 GitHub

hostingowy serwis internetowy przeznaczony dla projektów programistycznych wykorzystujących system kontroli wersji Git. Stworzony został przy wykorzystaniu frameworka Ruby on Rails i języka Erlang. Serwis działa od kwietnia 2008 roku.  
W kwietniu 2011 ogłoszono, iż GitHub obsługuje 2 miliony repozytoriów. GitHub udostępnia darmowy hosting programów open source oraz płatne prywatne repozytoria, został wybrany ze względu na możliwość tworzenia darmowego repozytorium w przypadku wersji open source oraz możliwość przechowywania repozytorium w chmurze

## 2.2 Slack

Darmowa usługa internetowa oparta na chmurze zawierająca zestaw narzędzi i usług służących współpracy zespołowejstworzona w oparciu o platformę programistyczną Electron. Aplikacja pełni rolę komunikatora internetowego umożliwiającego komunikację tekstową oraz głosową wraz z możliwością wysyłania multimediów.

## 2.3 Uzasadnienie wybranych narzędzi

### 2.3.1:GitHub

Program GitHub został wybrany ze względu na możliwość tworzenia darmowych repozytoriów typu open source w chmurze, dodatkowo istnieje możliwość tworzenia prywatnych repozytoriów w opraciu o studencką licencję jednakże zdecydowaliśmy się nie iść w tym kierunku tak by inne osoby mogły wzorować się na naszym projekcie

### 2.3.2:Slack

Program slack został wykorzystany ze względu na to że jest on bardzo praktycznym narzędziem, które dodatkowo w wersji darmowej nie przechowuje danych na serwerze co sprawia że jest bardziej prywatny. Dodatkowo w przeciwieństwie do najbardziej znanych globalnych komunikatorów („Messenger”) istnieje możliwość skupienia się na pracy bez ingerencji osób trzecich.

### 2.3.3: Inne narzędzia

Ze względu na mały zespół tworzący projekt zrezygnowaliśmy z narzędzi służących do kontroli postępów pracy typu jira lub hack&plan. Cele oraz terminy ustaliliśmy podczas pierwszego spotkania i staraliśmy się ich trzymać, w przypadku ewentualnych opóźnień szczegóły podajemy za pośrednictwem slacka

# 3.Testy:

Testy wykonujemy dwuetapowo

## 3.1: Testy integracyjne

Testy integracyjne z użyciem „nose”, które pokrywa podstawowe funkcjonalności



## 3.1: Testy manualne

Testy zostały wykonane na dwóch komputerach w sieci lokalnej

# Źródła:

<https://docs.python.org/3/>  
(dostęp 02.01.2019) Oficjalna dokumentacja Pythona  
<https://docs.python.org/2/library/difflib.html>  
(dostęp 02.01.2019) Oficjalna dokumentacja Python - klasa difflib  
<http://www.metal.agh.edu.pl/~banas/SRR/SRR_W09_RPC_RMI.pdf>  
(dostęp 02.01.2019)  Dr hab. inż. Krzysztof Banaś, Akademia Górniczo Hutnicza w Krakowie  
<https://www.python.org/download/releases/3.0/>  
(dostęp 02.01.2019) Python 3.0 Release – source  
<http://nose.readthedocs.io/en/latest/>  
(dostęp 02.01.2019) Nose - python tests