



Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie

Wydział Inżynierii Mechanicznej i Robotyki

TECHNOLOGIA MOWY

Projekt: Osobisty Organizator Głosowy

Skład zespołu: Magdalena Pawlak, Olga Krzyżyńska, Jadwiga Michalik

Rok akademicki: 2020/2021

CEL PROJEKTU

Projekt ma charakter wdrożeniowy. Jego celem jest realizacja systemu pomagającego w organizacji czasu. System ten umożliwia użytkownikowi głosowe obsługiwanie internetowego kalendarza (w naszym przypadku Kalendarza Google). Program dodaje nowe wydarzenia oraz wyświetla i odczytuje wydarzenia istniejące.

ZAŁOŻENIA PROJEKTU

Nasz organizer współpracuje z internetowym Kalendarzem Google, dlatego niezbędnym jest posiadanie konta Google. System przed rozpoczęciem głosowej obsługi organizera wymaga wykonania przez użytkownika następujących czynności:

- ★ zainstalowania interpretera i pakietów zawartych w pliku "requirements.txt", dostępnego pod linkiem: <https://github.com/gosc32123/Osobisty-Organizer-Glosowy>
- ★ zapisania pliku 'credentials.json' udostępnionego przez autorów projektu, który zawiera w sobie dane dostępowe do konta Google; plik został wygenerowany za pomocą Google Cloud API
- ★ ręcznej autoryzacji konta Google podczas pierwszego uruchomienia
- ★ zapisania pliku 'słownik.pkl', czyli zserializowanego słownika potrzebnego do obsługi systemu (zawartość słownika została opisana dokładnie niżej)

OPIS ARCHITEKTURY I UŻYTYCH SYSTEMÓW

★ Użyte biblioteki i systemy

- określenie obecnej daty
 - 'datetime'
- serializacja słownika
 - 'pickle'
- instalacja Kalendarza Google
 - '__future__'
 - 'googleapiclient.discovery'
 - 'google_auth_oauthlib.flow'
 - 'google.auth.transport.requests'
- rozpoznawanie mowy (ASR)
 - Google Speech Recognition ('speech_recognition')
- odczytywanie i odtwarzanie mowy (TTS)
 - Google TTS ('gtts')
 - 'playsound'
- operowanie na plikach i folderach
 - 'os'
- obsługa systemu
 - 'sys'

- ★ **Słownik** umożliwiający przypisanie konkretnego klucza do tego, co powie użytkownik niezależnie od sposobu wypowiedzenia (odmiany). Klucze posiadają format akceptowany przez Kalendarz Google. Słownik został oryginalnie utworzony na potrzeby projektu.

```
dict = {
    "00:00": ['zero', 'północ', 'północy', 'dwudziesta czwarta', '0:00', '24'],
    "1": ['jeden', 'pierwszy', 'pierwszego', 'styczeń', 'pierwsza', 'stycznia', '1:00'],
    "2": ['drugi', 'dwa', 'drugiego', 'luty', 'druga', 'lutego', '2:00'],
    "3": ['trzeci', 'trzeciego', 'trzy', 'marzec', 'trzecia', 'marca', '3:00'],
    "4": ['cztery', 'czwartego', 'czwarty', 'kwiecień', 'czwarta', 'kwietnia', '4:00'],
    "5": ['piąty', 'pięć', 'piątego', 'maj', 'piąta', 'maja', 'my', '5:00'],
    "6": ['sześć', 'szóstego', 'szósty', 'czerwiec', 'szósta', 'czerwca', '6:00'],
    "7": ['siedem', 'siódmego', 'siódmy', 'lipiec', 'siódma', 'lipca', '7:00'],
    "8": ['osiem', 'ósmego', 'ósmy', 'sierpień', 'ósma', 'sierpnia', '8:00'],
    "9": ['dziewięć', 'dziewiąty', 'dziewiątego', 'wrzesień', 'dziewiąta', 'września', '9:00'],
    "10": ['dziesięć', 'dziesiątego', 'dziesiąty', 'październik', 'dziesiąta', 'października', '10:00'],
    "11": ['jedenasty', 'jedenastego', 'jedenaście', 'listopad', 'jedenasta', 'listopada', '11:00'],
    "12": ['dwunasty', 'dwunastego', 'dwanaście', 'grudzień', 'dwunasta', 'grudnia', '12:00'],
    "13": ['trzynaście', 'trzynastego', 'trzynasty', 'trzynasta', '13:00'],
    "14": ['czternaście', 'czternastego', 'czternasty', 'czternasta', '14:00'],
    "15": ['piętnaście', 'piętnastego', 'piętnasty', 'piętnasta', '15:00'],
    "16": ['szesnaście', 'szesnastego', 'szesnasty', 'szesnasta', '16:00'],
    "17": ['siedemnaście', 'siedemnastego', 'siedemnasty', 'siedemnasta', '17:00'],
    "18": ['osiemnaście', 'osiemnastego', 'osiemnasty', 'osiemnasta', '18:00'],
    "19": ['dziewiętnaście', 'dziewiętnastego', 'dziewiętnasty', 'dziewiętnasta', '19:00'],
    "20": ['dwudziesty', 'dwudziestego', 'dwudziesty', 'dwudziesta', '20:00'],
    "21": ['dwudziesty pierwszy', 'dwudziestego pierwszego', 'dwadzieścia jeden', 'dwudziesta pierwsza', '21:00'],
    "22": ['dwudziesty drugi', 'dwudziestego drugiego', 'dwadzieścia dwa', 'dwudziesta druga', '22:00'],
    "23": ['dwudziesty trzeci', 'dwudziestego trzeciego', 'dwadzieścia trzy', 'dwudziesta trzecia', '23:00'],
    "24": ['dwudziesty czwarty', 'dwudziestego czwartego', 'dwadzieścia cztery'],
    "25": ['dwudziesty piąty', 'dwudziestego piątego', 'dwadzieścia pięć'],
    "26": ['dwudziesty szósty', 'dwudziestego szóstego', 'dwadzieścia sześć'],
    "27": ['dwudziesty siódmy', 'dwudziestego siódmego', 'dwadzieścia siedem'],
    "28": ['dwudziesty ósmy', 'dwudziestego ósmego', 'dwadzieścia osiem'],
    "29": ['dwudziesty dziewiąty', 'dwudziestego dziewiątego', 'dwadzieścia dziewięć'],
    "30": ['trzydziesty', 'trzydziestego', 'trzydzieści'],
    "31": ['trzydziesty pierwszy', 'trzydziestego pierwszego', 'trzydzieści jeden'],
    "32": ['trzydzieści dwa'],
    "33": ['trzydzieści trzy'],
    "34": ['trzydzieści cztery'],
    "35": ['trzydzieści pięć'],
    "36": ['trzydzieści sześć'],
    "37": ['trzydzieści siedem'],
    "38": ['trzydzieści osiem'],
    "39": ['trzydzieści dziewięć'],
    "40": ['czterdzieści'],
    "41": ['czterdzieści jeden'],
    "42": ['czterdzieści dwa'],
    "43": ['czterdzieści trzy'],
    "44": ['czterdzieści cztery'],
    "45": ['czterdzieści pięć'],
    "46": ['czterdzieści sześć'],
    "47": ['czterdzieści siedem'],
    "48": ['czterdzieści osiem'],
    "49": ['czterdzieści dziewięć'],
    "50": ['pięćdziesiąt'],
    "51": ['pięćdziesiąt jeden'],
    "52": ['pięćdziesiąt dwa'],
    "53": ['pięćdziesiąt trzy'],
    "54": ['pięćdziesiąt cztery'],
    "55": ['pięćdziesiąt pięć'],
    "56": ['pięćdziesiąt sześć'],
    "57": ['pięćdziesiąt siedem'],
    "58": ['pięćdziesiąt osiem'],
    "59": ['pięćdziesiąt dziewięć'],
    "DAILY": ['codziennie', 'co dzień', 'każdego dnia'],
    "WEEKLY": ['co tydzień'],
    "MONTHLY": ['co miesiąc'],
    "YEARLY": ['co rok', 'co roku'],
    "email": ['mailem', 'mailowo', 'mail', 'email', 'email'em', 'wiadomością', 'wiadomość'],
    "popup": ['powiadomienie']
}
```

Rys.1. Słownik współpracujący z systemem.

SKRÓTOWA DOKUMENTACJA SYSTEMU

Kod całego systemu dostępny jest na [githubie](#).

★ Funkcje:

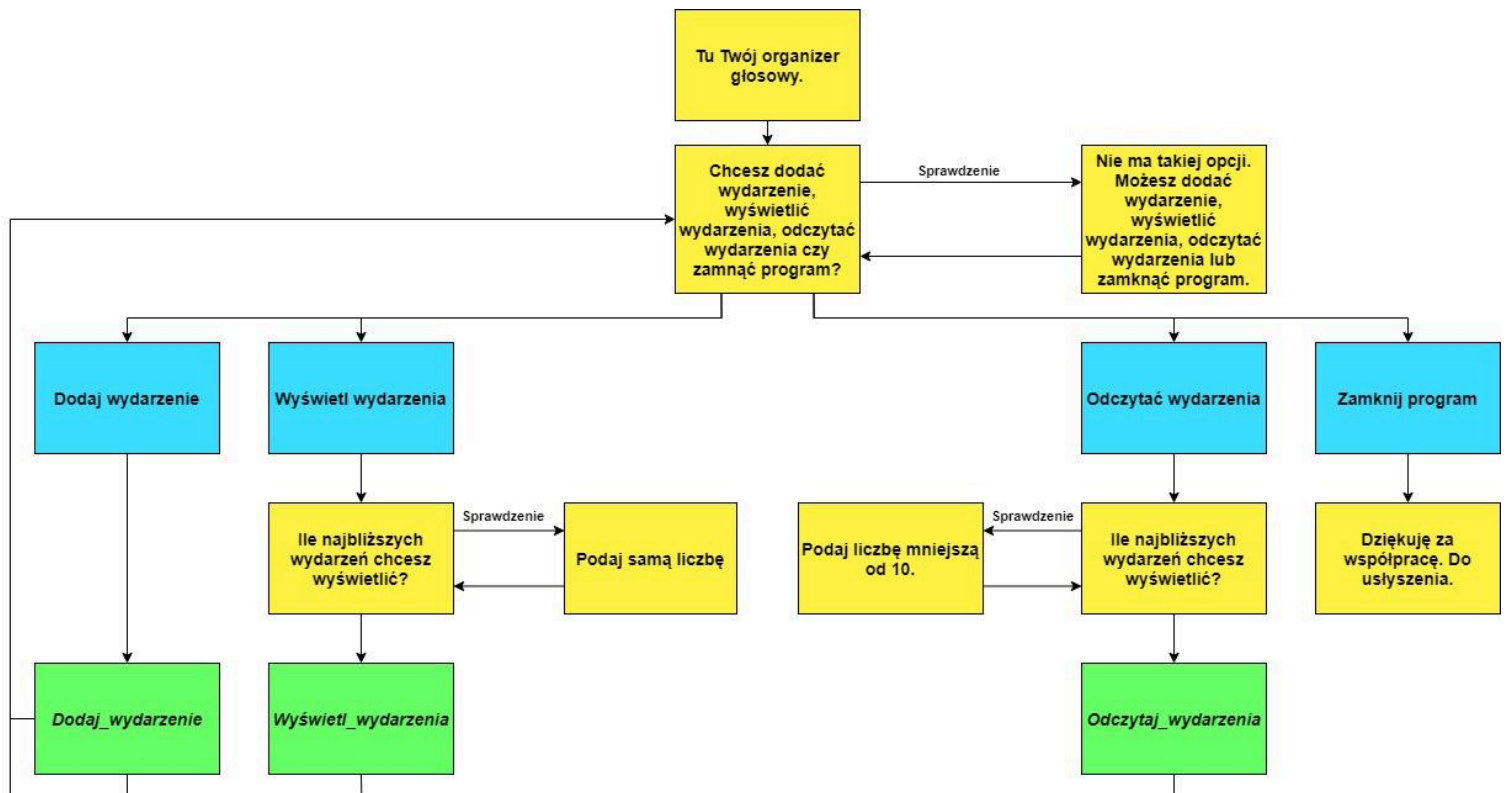
- 'rozpoznaj' - zamienia tekst mówiony na pisany (ASR)
- 'powiedz' - zamienia tekst pisany na mówiony (TTS)
- 'str_to_num' - zamienia liczby pisane słownie na liczby pisane numerycznie (za pomocą słownika)
- 'dni_tygodnia' - zwraca datę na podstawie dni tygodnia
- 'godziny' - zwraca godzinę
- 'minuty' - zwraca minuty
- 'czestosc' - zwraca jak często ma się powtarzać wydarzenie ('codziennie', 'co tydzień', 'co miesiąc', 'co rok')
- 'ilosc' - zwraca ile razy ma się powtarzać wydarzenie (wartość liczbową)
- 'metoda' - zwraca sposób przypomnienia ('powiadomienie', 'email')
- 'data' - korzysta z funkcji 'dni_tygodnia' oraz ze słownika; zwraca datę
- 'poprawa' - służy do poprawienia błędnych danych
- 'dodaj_do_kalendarza' - dodaje wydarzenie do Kalendarza Google
- 'dodaj_wydarzenie' - przeprowadza dialog z użytkownikiem, korzysta ze wszystkich powyższych funkcji
- 'wyswietl' - służy do wyświetlania wydarzeń dostępnych w kalendarzu
- 'odczytywanie' - służy do odczytywania wydarzeń dostępnych w kalendarzu
- 'interfejs' - menu początkowe

★ **Wyjątki i zabezpieczenia** zawarte w systemie, które zapobiegają błędnym komunikatom i umożliwiają różne sposoby podawania danych. Poniżej znajduje się lista wyjątków i zabezpieczeń uwzględnionych w systemie.

- przed wcześniejszą datą zakończenia niż rozpoczęcia
- przed podaniem niepoprawnych danych (np. zły format daty lub godziny)
- przed podaniem niedostępnych opcji (np. 'powiadomienie SMS-em')
- różne sposoby podawania godziny
 - sama godzina (np. 'czternasta', '14:00')
 - godzina z minutami (np. '14:35')
 - ile minut przed (np. '5 minut przed 14:00')
 - ile minut po (np. '5 minut po')

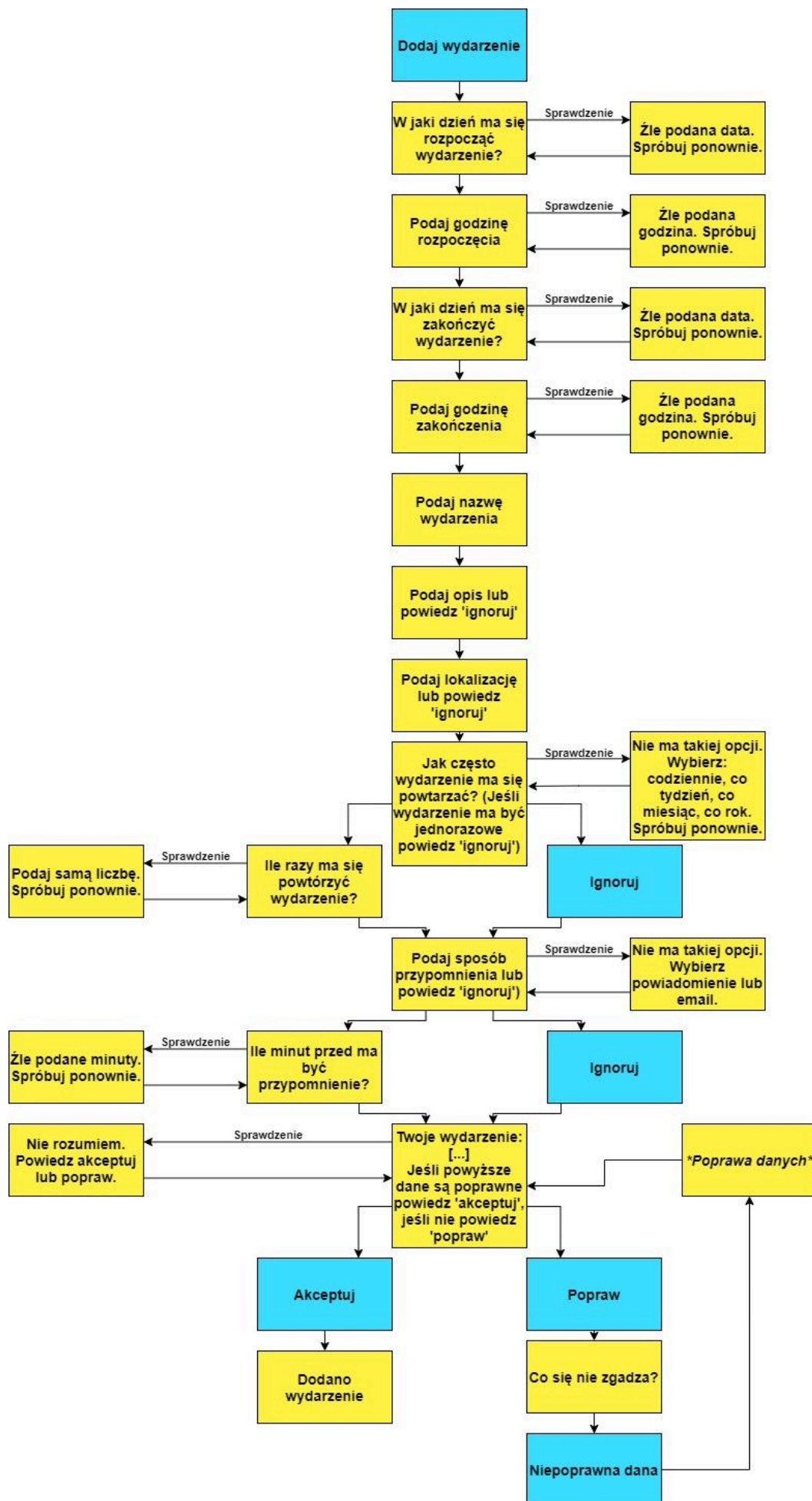
- 'kwadrans po', 'kwadrans przed'
- 'w pół do'
- różne sposoby podawania daty
 - 'dzisiaj'
 - 'jutro'
 - 'pojutrze'
 - konkretny dzień w obecnym tygodniu (np. 'w ten poniedziałek')
 - konkretny dzień w kolejnym tygodniu (np. 'w przyszły poniedziałek')
 - data 'dzień miesiąc' (np. '27 stycznia')
 - data 'dzień miesiąc rok' (np. '27 stycznia 2023 roku')
 - data zakończenia 'w ten sam dzień', 'tego samego dnia'

★ Menu początkowe



Rys. 2. Schemat działania menu początkowego.

★ Dialog do dodawania wydarzeń



Rys. 3. Schemat dialogu systemu z użytkownikiem.

OSIĄGNIĘTE SUKCESY

- ★ system “rozmawia” z użytkownikiem, komendy i pytania są odczytywane głosowo za pomocą Google TTS, a mowa użytkownika rozpoznawana za pomocą Google Speech Recognition
- ★ system łączy się z Kalendarzem Google, dodaje do niego wydarzenia, a także je wyświetla i odczytuje
- ★ system akceptuje różne sposoby podawania informacji przez użytkownika

TESTOWANIE SYSTEMU

System testowany był przez osoby niezwiązane z projektem, w tym osoby nieznające sposobu działania Kalendarza Google. W sumie było kilkunastu niezależnych testujących, którzy przeprowadzili testy po kilka razy. Każdy testujący miał dostęp do głównego programu (‘py’), słownika (‘pkl’) oraz pliku autoryzującego (‘credentials.json’). Przed rozpoczęciem testu testerzy wiedzieli tylko, że mają do czynienia z organizerem głosowym, który dodaje wydarzenia do Kalendarza Google, nie znali dialogu i nie wiedzieli, o co zapyta system. Przeprowadzone testy pomogły nam udoskonalić drzewo dialogowe i wskazały, które pytania są niejasne. Pokazały też, jakie wyjątki powinniśmy dodać (wszystkie wyjątki zostały opisane powyżej w pkt. Wyjątki i zabezpieczenia).

NAPOTKANE PROBLEMY I ICH ROZWIĄZANIE

- ★ błędne rozpoznawanie niektórych słów użytkownika przez system
(np. ‘czy’ zamiast ‘trzy’, ‘pomień’ zamiast ‘pomiń’, ‘my’ zamiast ‘maj’)

Rozwiązanie: testowanie systemu i odnajdowanie problematycznych słów, a następnie przypisywanie im odpowiednich kluczy w słowniku. W przypadku niektórych wyrazów wymagana była modyfikacja dialogu, tak by nie były one używane (np. zamiana komendy ‘pomiń’ na ‘ignoruj’).

- ★ programowanie za pomocą wyrażeń regularnych i wyjątków

Ten sposób tworzenia programu uniemożliwia przygotowanie systemu na wszystkie możliwe odpowiedzi użytkownika. Został wykorzystany, bo celem projektu było osiągnięcie szybkich efektów.

Rozwiązanie: alternatywą dla programowania za pomocą wyjątków może być blok NLP (Neuro-Linguistic Programming), który często wykorzystywany jest razem z systemami ASR. Metoda ta zajmuje się przetwarzaniem języka naturalnego, nie korzysta z programowania za pomocą wyjątków.

★ ułożenie prawidłowego dialogu

Problemem były zbyt ogólne pytania lub takie, które zakładały wiedzę użytkownika na temat działania programu. Kłopotliwe okazały się także pytania z odpowiedziami 'Tak' lub 'Nie'.

Rozwiązanie: testowanie systemu na osobach niezwiązanych z projektem i modyfikacja dialogu (na pytania sugerujące).

★ czasochłonne dodawanie wydarzeń

Pierwotnym przeznaczeniem lektora Google TTS była współpraca z tłumaczem, dlatego tempo wypowiadania przez niego słów jest dość wolne. Wydłuża to pracę systemu.

Rozwiązanie: w przyszłości można przyspieszyć tempo wypowiadania słów przez lektora lub wykorzystać inny system do zamiany tekstu na mowę.

POMYSŁY NA DALSZY ROZWÓJ PROJEKTU

Aktualna wersja systemu jest bardzo użyteczna i pomocna w codziennym używaniu Kalendarza Google. Jednak jest wiele rzeczy, które mogą jeszcze bardziej usprawnić jego działanie.

W przyszłości warto:

- ★ dodać możliwość edytowania oraz usuwania istniejących wydarzeń
- ★ dopracować sposób wyświetlania wydarzeń (np. przekierowanie do Kalendarza Google)
- ★ stworzyć aplikację, aby system był używalny niezależnie od posiadania środowiska programistycznego, które obsługuje język Python
- ★ spróbować użyć wyżej wspomnianego bloku NLP zamiast wyjątków

WNIOSKI Z PROJEKTU

Założony cel projektu został osiągnięty, ponieważ udało nam się stworzyć w pełni działający system, który ma potencjał na dalszy rozwój. Praca nad projektem przyniosła nam wiele nowych doświadczeń. Dzięki niej poznaliśmy zasadę działania systemów głosowych, których tak często używamy na co dzień. Zaobserwowaliśmy, skąd wynikają błędy w rozumieniu mowy przez systemy ASR (np. przez zbyt podobne do siebie słowa lub niedopracowany system). Podczas pracy rozwinęliśmy również nasze umiejętności programistyczne, poznaliśmy nowe biblioteki i funkcje w języku Python oraz udoskonaliłyśmy umiejętność pracy w zespole. Projekt był czasochłonny, ale bardzo satysfakcjonujący. Dzięki wykonanej pracy możemy teraz wykorzystywać stworzony przez nas Organizator Głosowy do codziennej obsługi Kalendarza Google.

Wyrażamy zgodę na prezentację wyników projektu, którego dotyczy ten raport na stronie Zespołu Przetwarzania Sygnałów oraz w ramach wydarzeń promujących uczelnię, takich jak Noc Naukowców, Dzień Otwarty AGH, Festiwal Nauki.

Wyrażamy zgodę na publikację naszych imion i nazwisk podczas prezentacji wyników projektu, którego dotyczy ten raport.