

Jakub Korzeniowski

45432

Informatyka, sem III

Pl.mail

skrypt do prostej obsługi gmaila oraz automatyzacji informacji

SPIS TREŚCI

1. Wstęp:
 - a. O aplikacji
2. Potrzeba stworzenia/geneza problemu/dlaczego brak GUI
3. Analiza popytu
4. Studium wykonalności
5. Przygotowanie urządzenia
6. Budowa skryptu
 - a. Możliwości
 - i. Wyślij
 - ii. Odbierz
 - iii. Dołącz/pobierz załącznik
7. Generacja kluczy OAuth2 oraz token'u gmail
8. Inicjalizacja skryptu
 - a. Połączenie konta gmail ze skryptem
9. Działanie skryptu
10. Użycie skryptu
 - a. Diagram przypadków użycia
11. Wymagania funkcjonalne i нефункционалне
12. Podsumowanie

Wstęp

O aplikacji:

Aplikacja jest stworzona w formie skryptu w języku Python. Połączenie z usługą Gmail zapewnione jest dzięki dostępnemu API zapewnionego przez Gmail dla developer'ów. Aplikacja zapewnia takie funkcje jak: wysłanie maila do danego odbiorcy z wybranym tematem, podanie treści maila oraz dołączenie załącznika. Drugą funkcją jest możliwość odczytania wiadomości email. Skrypt sam wykrywa czy do wiadomości dodany jest załącznik – jeśli tak, prosi o podanie ścieżki do folderu, w którym ów załącznik ma zostać zapisany, jeśli nie, kończy działanie skryptu wraz z jego odczytaniem. Ostatnią funkcją jest sprawdzenie, z jakiego adresu email właśnie korzystamy, ze względu na brak GUI mogą występować pomyłki, dlatego opcja sprawdzenia swojej aktualnej „tożsamości” jest ważną funkcją.

Dalsze plany rozwoju projektu zakładają możliwość automatyzacji wysyłki maili na bazie pliku excel. Aby mail został wysłany do odpowiednich osób każdy rekord z emailem musi zawierać atrybut wyróżniający go, taki jak True or False, by został wykryty przez skrypt.

Przykładowy scenariusz:

Pracownik działu HR musi wysłać maila do pracowników z informacją o problemie z księgowością, przez co część pracowników nie otrzymała wypłaty. Dane dotyczące maili oraz tego, czy pracownik otrzymał wypłatę znajduje się w pliku Excel posiadającym 1500 rzędów, oraz 2 kolumny: Email oraz „Czy otrzymał wypłatę”. Weryfikacja osób oraz wysłanie maila do każdego z osobna byłoby bardzo czasochłonne i nieefektywne. Skrypt wykonuje tą pracę w bardzo krótkim czasie.

Jest to plan na zapewnienie automatyzacji informacji na dużą skalę.

Potrzeba utworzenia

Geneza problemu:

Wraz z postępem i rozwojem technologii, pojawia się coraz więcej opcji i funkcji, z którym można korzystać. Ze względu na to rozwija się ilość elementów wyświetlanych w GUI użytkownika powodując mniejszą czytelność. Dla osób młodych, szybko rozwijających się nie stanowi to problemu, każda funkcja wydaje się być intuicyjna. Natomiast osoby starsze często borykają się z problemem „nie nadążania” za technologią.

Stąd właśnie BRAK GUI. Skrypt wykonywany jest w terminalu – interfejs tekstowy. Program zawiera zaledwie 3 podstawowe opcje: wyślij, odbierz, sprawdź dane – wszystko czego potrzebuje osoba starsza używająca mail’a. To założenie ma zapewnić prostotę używania skrzynki elektronicznej każdej osobie.

Kwestia automatyzacji jest podstawowym zagadnieniem każdej firmy pracującej z klientami fizycznymi. Przy dużej ilości odbiorców jednej wiadomości skrypt pozwala na automatyczne wysłanie do wszystkich, bądź wybranych osób wiadomości email.

Analiza popytu

1. Analiza interesariuszy i użytkowników:

- a. Usługi:
 - i. Wysyłanie/Odbieranie wiadomości email
 - ii. Dołączanie/Pobieranie załączników
 - iii. Wysyłanie szablonu maila do wielu użytkowników z bazy – excel
- b. Interesariusze:
 - i. Firmy/korporacje
 - ii. Reklamodawcy
- c. Użytkownicy:
 - i. Osoby starsze
 - ii. Osoby młodsze
 - iii. Pracownicy firm
 - iv. Agencje reklamowe

2. Popyt bieżący

Wg. Statystyk w marcu 2020 r. z usług Gmail'a skorzystało 11,5 mln użytkowników. W samej Polsce w roku 2020 osób powyżej 60 roku życia było 25,6% ludności.

3. Analiza danych wewnętrznych:

- a. Ilość pobrań
- b. Ilość odwiedzających stronę z zamieszczonym skryptem

Źródła:

https://stat.gov.pl/files/gfx/portalinformacyjny/pl/defaultaktualnosci/6002/2/3/1/sytuacja_osob_starszych_w_polsce_w_2020_r.pdf strona 13

Studium wykonalności

1. Zakres:

Aplikacja będzie rozwijana w formie skryptu i będzie zawierała takie funkcje jak:

- Zsynchronizowanie poczty z aplikacją
- Wysyłanie maili z załącznikiem
- Odbieranie maili i pobieranie załącznika
- Sprawdzenie aktualnego użytkownika

2. Wykonalność techniczna:

Aplikacja będzie opracowana przy pomocy Pythona wraz z pakietami: ezgmail, pandas, openpyxl, xlrd.

3. Wykonalność finansowa:

Koszt utworzenia aplikacji wyniesie \$60 000, koszt obejmuje utworzenie, rozwój oraz utrzymanie aplikacji na okres 3 miesięcy.

4. Ograniczenia:

- Finansowe:
 - Brak, potrzebne pakiety oraz programy są darmowe
- Organizacyjne:
 - Ze względu na możliwość złego sformatowania tekstu mogą wystąpić błędy w wysłaniu maili.
- Techniczne:
 - Gmail może przestać udostępniać API,
 - Plik excel może być źle sformatowany
- Rynkowe:
 - Duże, podobny skrypt może zostać wykonany w innych technologiach
- Prawne:
 - Brak

5. Wniosek:

Rzeczywisty rozwój aplikacji jest możliwy. Zapewni prosty sposób obsługi skrzynki mailowej oraz ułatwi pracę przedsiębiorcy oraz pracownikom działów HR. Koszt utworzenia jak i utrzymania jest rozsądny, a wszelkie wymagania są osiągalne.

Przygotowanie urządzenia

Urządzenie, na którym uruchamiany będzie skrypt musi zostać wyposażone w:

1. Python3
 - a. Zależności:
 - i. Ezgmail (pakiet z predefiniowanymi funkcjami do obsługi API gmail),
 - ii. Openpyxl (pakiet służący do odczytywania plików excel/csv),
 - iii. Pandas (pakiet służący do analizy danych),
 - iv. Xlrd (pakiet do odczytu i formatowania danych z plików Excel)
 - b. Opcjonalnie:
 - i. Środowisko PyCharm

Wszelkie wymagane zależności należy zainstalować za pomocą PIP'u (preferred installer program), który domyślnie instalowany jest wraz z instalacją Pythona. Aby pobrać pakiet należy w wierszu poleceń środowiska programistycznego użyć komendy „pip install <nazwa_pakietu>”. W wierszu poleceń Windows przed komendą należy użyć przedrostka python/python3/py zależnie od używanej wersji Pythona.

Budowa skryptu

Podstawowa część skryptu bazuje na pakiecie ezgmail, zapewniającym połączenie z API gmail'a oraz proste zprebuildowane funkcje do wysyłania i odbierania maili.

```
import ezgmail
```

Funkcja wysyłająca obiera argumenty (tutaj dołączany jest załącznik):

```
>>> ezgmail.send('recipient@example.com', 'Subject line', 'Body of the email',  
cc='friend@example.com', bcc='otherfriend@example.com,someoneelse@example.com')
```

Funkcja odbierająca pobiera nieodczytane wiadomości:

```
# pobranie nieodczytanych maili  
unreadmails = ezgmail.unread(maxResults=100)
```

Skrypt automatycznie wykrywa czy do maila załączony jest załącznik:

```
if len(unreadmails[mailToRead-1].messages[0].attachments) >= 1: #sprawdzenie czy istnieje załącznik
```

Pobrane wiadomości mają swoje atrybuty: subject, body, timestamp, sender, recipient oraz attachment. Po nich możemy wypisywać poszczególne elementy wiadomości.

Wbudowana funkcja EMAIL_ADDRESS pozwala na wyświetlenie aktualnie używanego adresu mailowego

```
print(ezgmail.EMAIL_ADDRESS)
```

(część skryptu odpowiadająca za wysyłanie maili na podstawie pliku excel jest w trakcie tworzenia)

Generowanie klucza OAuth2

Oraz tokenu autoryzacji google

W celu utworzenia klucza autoryzacji OAuth2 użytkownik musi zalogować się do portalu deweloperskiego google pod linkiem

<https://developers.google.com/gmail/api/quickstart/python?hl=pl>.

Generacje klucza należy zacząć od włączenia interfejsu API,

Włącz API

Zanim zaczniesz korzystać z interfejsów API Google, musisz włączyć je w projekcie Google Cloud. W jednym projekcie Google Cloud możesz włączyć jeden lub więcej interfejsów API.

- W konsoli Google Cloud włącz interfejs Gmail API.

Włączanie interfejsu API

A następnie autoryzowania danych logowania

Autoryzowanie danych logowania do aplikacji komputerowej

Aby uwierzytelnić się jako użytkownik końcowy i uzyskać dostęp do danych użytkownika w aplikacji, musisz utworzyć co najmniej 1 identyfikator klienta OAuth 2.0. Identyfikator klienta służy do identyfikowania jednej aplikacji na serwerach OAuth Google. Jeśli aplikacja działa na wielu platformach, musisz utworzyć oddzielny identyfikator klienta dla każdej z nich.

1. W konsoli Google Cloud kliknij Menu ☰ > **Interfejsy API i usługi** > **Dane logowania**.

Otwórz stronę Dane logowania

2. Kliknij **Utwórz dane logowania** > **Identyfikator klienta OAuth**.
3. Kliknij **Typ aplikacji** > **Aplikacja komputerowa**.
4. W polu **Nazwa** wpisz nazwę danych logowania. Ta nazwa jest widoczna tylko w konsoli Google Cloud.
5. Kliknij **Utwórz**. Pojawi się ekran utworzony przez klienta OAuth zawierający nowy identyfikator klienta i tajny klucz klienta.
6. Kliknij **OK**. Nowo utworzone dane logowania pojawią się w sekcji **Identyfikatory klienta OAuth 2.0**.
7. Zapisz pobrany plik JSON jako `credentials.json` i przenieś go do katalogu roboczego.

Wykonanie powyższych kroków skutkuje zautoryzowaniem klienta.

Inicjalizacja skryptu

W celu zainicjowania skryptu po instalacji wszystkich zależności w konsoli należy przejść do katalogu w którym znajdują się nasz skrypt.

```
C:\Users\kubak>cd PycharmProjects\ProjectMail

C:\Users\kubak\PycharmProjects\ProjectMail>dir
Volume in drive C has no label.
Volume Serial Number is 8E92-DD90

Directory of C:\Users\kubak\PycharmProjects\ProjectMail

22.01.2023  18:29    <DIR>          .
22.01.2023  18:29    <DIR>          ..
29.01.2023  14:39    <DIR>          .idea
22.01.2023  18:29    <DIR>          build
14.01.2023  14:41             399 credentials.json
22.01.2023  19:26    <DIR>          dist
22.01.2023  18:27             2 847 main.py
22.01.2023  19:26             936 main.spec
24.01.2023  14:01             1 228 token.json
               4 File(s)              5 410 bytes
               5 Dir(s)  15 191 048 192 bytes free

C:\Users\kubak\PycharmProjects\ProjectMail>
```

Zainicjowanie skryptu komendą „py main.py” przeniesie nas do przeglądarki w celu zalogowania do konta google przypisanego jako klient projektu z kluczem OAuth2.

```
C:\Users\kubak\PycharmProjects\ProjectMail>py main.py

Your browser has been opened to visit:

  https://accounts.google.com/o/oauth2/

If your browser is on a different machine then exit and re-run this
application with the command-line parameter

  --noauth_local_webserver
```

```
Authentication successful.
←[91mUnread emails stack at 100 positions←[0m
←[92mYou have 5 unread mails←[0m
1. Send email
2. Read email
3. what's my mail
←[92mwhat do you want to do?: ←[0m3
projektindtest@gmail.com
```

Przypadki użycia

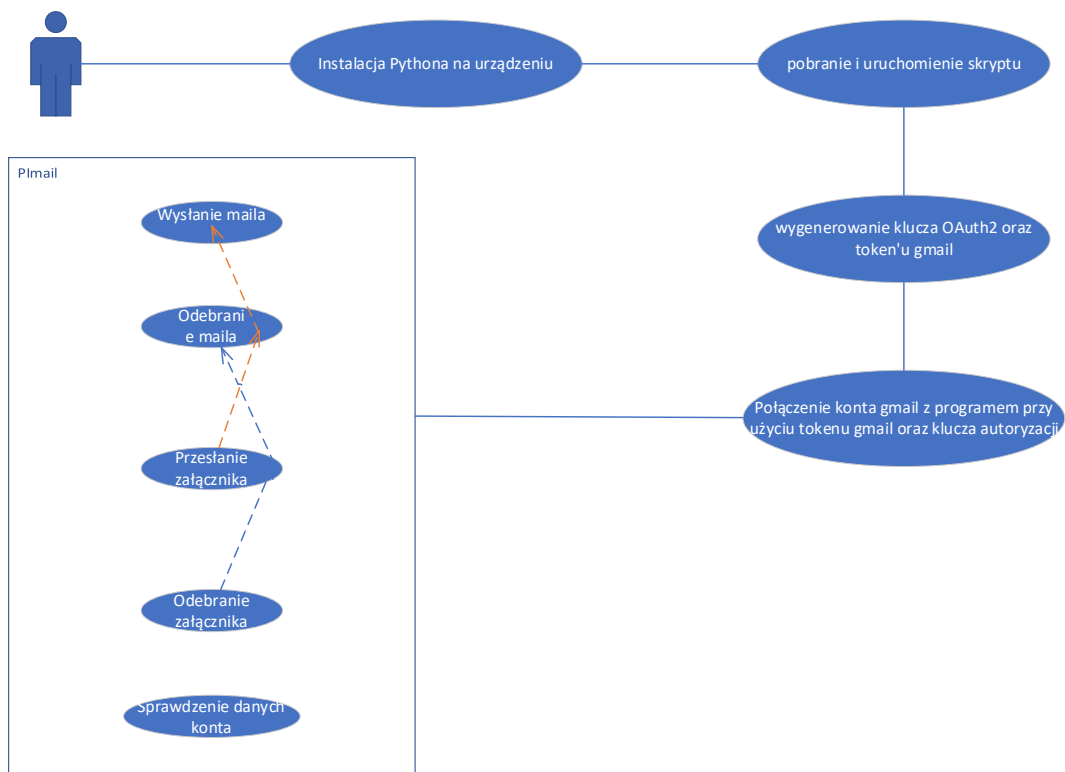


Diagram 1. Instalacja/konfiguracja

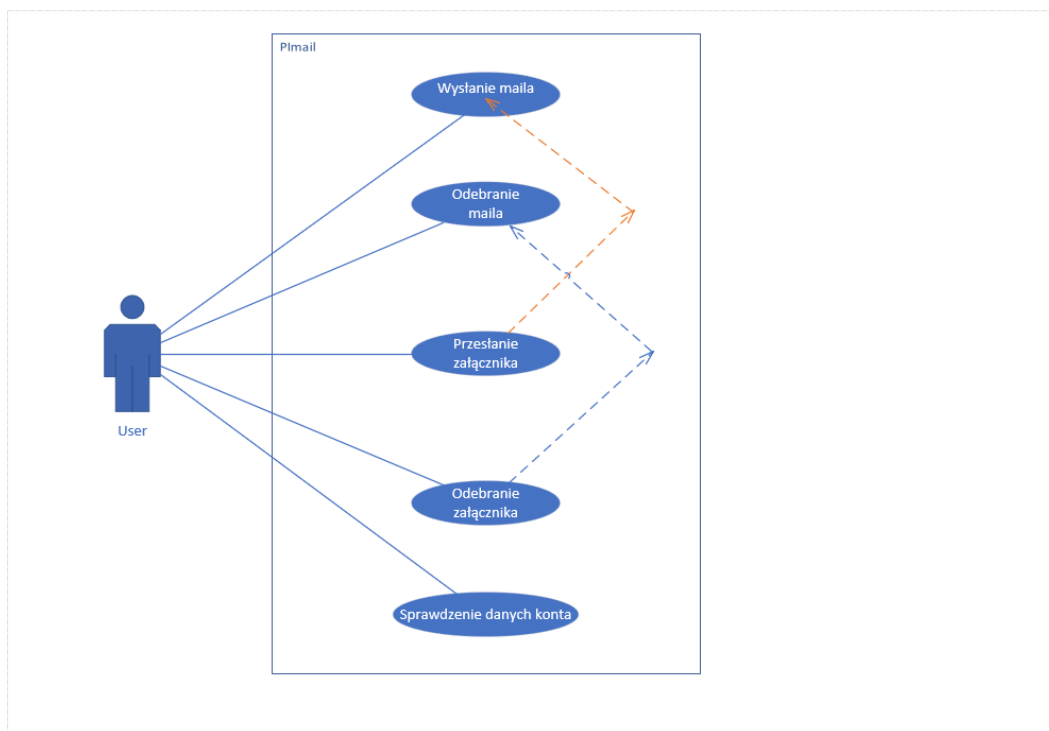


Diagram 2. Przypadki użycia skryptu

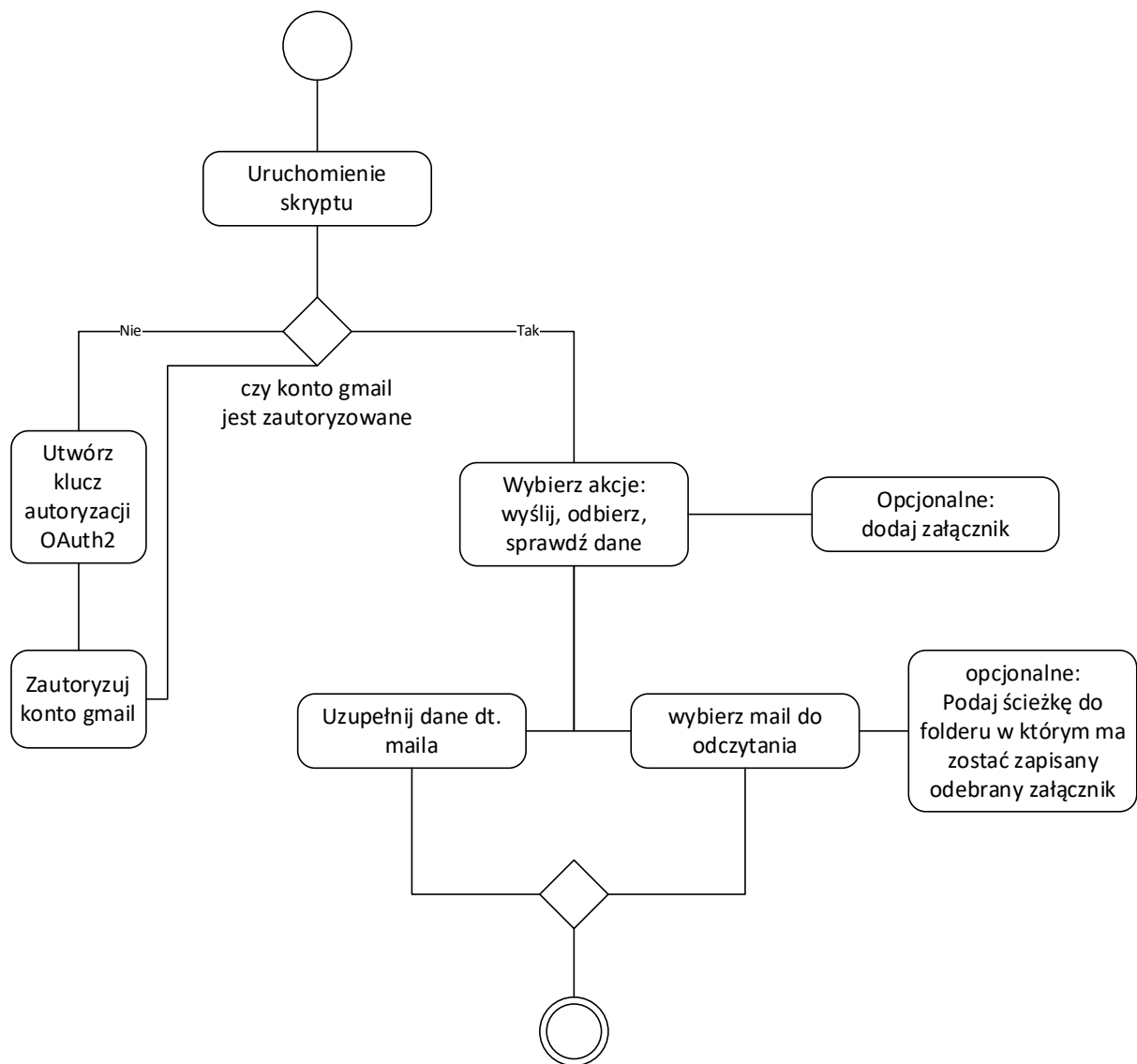


Diagram 3. UML activity

Wymagania (nie)funkcjonalne

Funkcjonalne:

- Możliwość wysyłania i odbierania maili
- Możliwość zapisywania załączników
- Możliwość dodawania filtrów do wiadomości
- Możliwość wysyłania maili zbiorowych do kontaktów zapisanych w pliku excel bazujących na danym atrybucie
- Możliwość wyszukiwania wiadomości po frazie
- Możliwość wyświetlania powiadomień o nowych wiadomościach

Niefunkcjonalnie:

- Łatwa nawigacja użytkownika
- Szybkie wczytywanie danych
- Prosty w obsłudze
- Łatwa integracja z usługą Gmail
- Możliwość szybkiego wyszukiwania wiadomości
- Szyfrowanie danych użytkownika w kluczu OAuth2

Podsumowanie

Projekt PImail jest prostym w obsłudze skryptem do obsługi maila przeznaczonym dla osób, które nie potrzebują żadnych opcji dodatkowych poza wysyłaniem i odbieraniem maili. Druga część przeznaczona jest dla firm i przedsiębiorstw potrzebujących zautomatyzowania procesu wysyłania maili masowych. Wykonanie projektu nie jest kosztochłonne, przez co cieszyć może się sporym zainteresowaniem.