Univerzitet u Beogradu

Fakultet organizacionih nauka

Katedra za elektonsko poslovanje

TEMA SEMINARSKOG RADA

Seminarski rad iz Elektronskog poslovanja

Nastavnik: Prof. dr Zorica Bogdanović Studenti:   
Saradnik: Tamara Naumović Aleksa Mirković 2020/1011

Andrea Filipović 2019/0292

Lena Lončarević 2019/0017

Beograd, 2022.

Sadržaj

[**1.** **UVOD** 3](#_Toc94465798)

[**2.** **PREGLED LITERATURE** 4](#_Toc94465799)

[**3.** **PREGLED POSTOJEĆIH REŠENJA** 5](#_Toc94465800)

[**4.** **RAZVOJ APLIKACIJE** 6](#_Toc94465801)

[**5.** **ZAKLJUČAK** 10](#_Toc94465802)

[**6.** **KORIŠĆENA I REFERENTNA LITERATURA** 11](#_Toc94465803)

# **UVOD**

Ne tako davne 2008. godine na filmskoj sceni pojavljuje se film “Iron Man”. Naš kolega Aleksa Mirković od prvog trenutka postao je opsednut filmom. Upravo iz tog razloga mi smo obradili temu Python I Google Speech inspirisani likom Jarvis-a, Iron Man-ovog inteligentnog sistema pomoćnika, kao veliku zelju našeg kolege.

U drugom poglavlju rada navodimo i približavamo korišćenu literaturu, odnosno eksterne biblioteke koje su nam bile potrebne za izradu pogramskog koda.

U trećem poglavlju predstavljamo pregled već postojećih popularnih aplikacija koje nudi slične ili iste funcionalnosti kao naša aplikacija poput svima poznatih “voice assistant”-a Siri ili Alex-e.

U četvrtom poglavlju objašnjavamo tok razvoja naše aplikacije. Osnovne funkcionalnosti možemo podeliti na tri celine:

1. Predstavljanje samog sebe

2. Otvaranje različitih aplikacija

3. Izračunavanje matematičkih kalkulacija

Naš “voice assistant” Katarina najpre će Vam se predstaviti svojim imenom, a potom upitati za Vaše, kao i način na koji može da Vam pomogne. Katarina može da računa komplikovana matematičke jednačine u par trenutaka, poznaje imena svojih kreatora i u sekundi može otvoriti neku aplikaciju koju joj naredite.

Mogućnosti daljeg razvoja funkcionalnosti aplikacije su velike, te tako u bližoj budućnosti planiramo da Katarini obezbedimo i mogućnost “surfovanja internetom”.

# **PREGLED LITERATURE**

Python je jedan od najpopularnijih programskih jezika današnjice. Python je jako koristina jezik za pisanje skripti a i za same developer-e zbog njegove jednostavnosti i razumljivosti. Samim tim postoje mnogo framework-a i biblioteka koje su napisane za razvoj aplikacija i za istraživanja.U našem projektu se sadrže nekoliko eksternih biblioteka (paketa) koje su nam pomogli u razvoju naše aplikacije:

1. gTTS-Google Text to Speech

2. speech recognition

3. wolframalpha

4. plasound

5. pyaudio

gTTS je eksterni paket koji je Google napravio za konvertovanje teksta u govor. Razvili su ga za potrebe njihovog ”voice assistant”-a koji je prisutan kod velikog broja Android uredaja. Zbog samog kvaliteta ovog paketa, on je samim tim defacto standard u svako aplikaciji koja koristi ovu tehnologiju.

speech-recognition je eksterni paket koji kako samo ime kaže, služi za konvertovanje zvukova u tekst. Nažalost, autor ovog paketa nije ostavio nikakvu dokumentaciju o ovom paketu, ali je jako koristan paket u kombinaciji sa gTTS.

wolframalpha je paket koji smo koristili za računsku funckionalnost naše aplikacije. Ovde smo koristili wolframalpha API i samim tim omogućili da naš ”voice assistant” može da reši razne matematičke probleme.

playsound je paket koji smo koristili za puštanje mp3 zvukova, s obzirom da naš program prvo sačuva našu zabeleženu komandu pa istu kasnije prosledi ostalim funkcijama.

pyaudio je paket sa kojim smo imali dosta problema. On ustvari predstavlja srž naše aplikacije jer je to voice engine koji upravlja glasom i izgovorom našeg asistenta.

# **PREGLED POSTOJEĆIH REŠENJA**

* **Siri** – razvili su programeri Iphone-a. To je asistent za oblak koji omogućava pokretanje programa, pretraživanje informacija i sveobuhvatno upravljanje mobilnim uredjajem uz pomoć glasovnih komandi
* **Google Now** (deo Android operativnog Sistema) – inteligentan agent koji nadogradjuje pretrage, odgovara na pitanja, pokreće aplikacije, odgovara na glasovne komande… On pokušava da predvidi potrebe i da pruži informacije pre nego što ih mi zatražimo.
* **Cortana** – Microsoftov virtuelni asistent. Dostupan je na Windows računarima, Android mobilnim telefonima i tabletima. Funkcioniše kao podsetnik, otvara aplikacije instalirane na računaru, prikazuje prognozu vremena, vrši proveru lokacije…
* **Alexa** – amazonski digitalni govorni asistent. Alexa Voice servis govorne signale transformiše u komande na računarskom jeziku koje mogu izvršiti zadatak (npr traženje odredjene pesme). Pored toga javlja se mogućnost pretvaranja u zvučne signale tako da potrebne informacije možemo da dobijemo verbalnim putem ( npr. o vremenu, saobraćaju…). Aparat može da se poveže sa aparatima koji su kompatibilni sa njim i da tako vrše neke funkcije kao što su pojačavanje ili smanjivanje zvuka, gašenje svetla…
* **Hound** – multiplatformni glasovni asistent koji je kreirao SoundHound. Od ostalih se razlikuje jer ne odvaja prepoznavanje govora i obradu jezika. Veoma je brz u obavljanju zadataka. Pruža mogućnost davanja uputa, prevodjenja tekstova, pretraživanja vesti, fotografija...
* **Google Home** – pametni zvučnici za dom. Uz pomoć njih možemo da puštamo muziku, regulišemo aparate po kući, dobijamo potrebne informacije…

1. **RAZVOJ APLIKACIJE**

Implementirani ”voice assistant” može da otvara instalirane aplikacije, da predstavi sam sebe i da reši razne matematičke probleme. Naravno ,uvek mogu da se dodaju funkcionalnosti, čak smo imali u planu da implementiramo da pretragu na Google i Youtube, ali jednostavno nismo imali vremena. Kao što smo napomenuli u 2. poglavlju, koristili smo nekoliko paketa u našoj implementaciji : gTTS , speech recognition , playsound , pyaudio , wolframalpha. Samim tim da bi korisnici mogli da koriste naˇsu aplikaciju moraju imati instaliran Python i date biblioteke. U tabeli ispod naći ćete komande kojim se instaliraju dati paketi. Komande ćete prekopirati u Command line ako ste u Windows operativnom sistemu, a u Terminal ako ste u Linux operativnom sistemu. Naravno podrazumeva se već imate Python interpretator na vašem sistemu.

|  |  |
| --- | --- |
| ***Paket*** | ***Komanda*** |
| ***gTTS*** | ***pip install gTTS*** |
| ***SpeechRecognition*** | ***pip install SpeechRecognition*** |
| ***wolframalpha*** | ***pip install wolframalpha*** |
| ***playsound*** | ***pip install playsound==1.2.2*** |
| ***pyaudio*** | ***pip install PyAudio*** |

Prvo smo krenuli sa krucijalnim implementacijama. To znači da smo prvo implementirali dve funkcije: funkciju speak koja kao output šalje govor samog asistenata i funkciju get audio koja prima audio sa mikrofona. U kodu ispod je implementacija ove dve funkcije sa main metodom.

Tabela Mirko pls ja nzm kako ovo samo font za ispod tabele ti je times new 10 bold I italic

import speech\_recognition as sr #ovo je za prepoznavanje govora

from gtts import gTTS # ovo je google API za pretvaranje teksta u govor import os #ovo je za cuvanje datoteka i otvaranje

import playsound #ovo je za pustanje zvukova koje imamo sacuvano na kompjuteru

from selenium import webdriver # za kontrolu browser-a

import wolframalpha # za racunski deo koda

num = 0

def speak(text): za\_govor = gTTS(text=text, lang=’en’)

filename = str(num)+".mp3"

za\_govor.save(filename)

playsound.playsound(filename)

os.remove(filename)

def get\_audio():

slusalac = sr.Recognizer()

with sr.Microphone() as source:

audio = slusalac.listen(source)

rekao = ""

try:

rekao = slusalac.recognize\_google(audio)

print(rekao)

except Exception as e:

print("GRESKA-> " + str(e))

return rekao

#MAIN FUNKCIJA

if \_\_name\_\_ == "\_\_main\_\_":

speak("What’s your name, Human?")

name =’Human’

name = get\_audio()

speak("Hello, " + name + ’.’)

while(1):

speak("What can i do for you?")

text = get\_audio().lower()

if text == 0:

continue

if "exit" in str(text) or "bye" in str(text) or "sleep" in str(text):

speak("Ok bye, "+ name+’.’)

break

process\_text(text)

Nakon što smo ovaj deo koda implementirali, sledeći korak je bio implementacija funkcionalnosti. shvatili smo da ne bi imalo smisla da u main funkciji imamo nekoliko if-a zbog estetike, pa smo sve to smestili u glavnu funkciju ovog koda process text. Unutar ove funkcije su sve funkcionalnosti koje smo naveli u početku. Unutar same funkcije videćete delove koda koji su pod komentarima. To smo ostavili tu tako da bi videli gde je trebala da bude implementacija pretrage na internetu. U delu koda za rešavanje matematičkih problema jako je bitno pomenuti da smo koristi wolframalpha API. Bilo je potrebno napraviti nalog na njihovom sajtu da bi se tek onda koristio njihov API. Kada se ulogujete, dobijate ID za vas projekat kako bi radila data funckionalnost. Taj ID smo kopirali u kod i napisali komande za slanje zahteva za računanje. Izraz koji želite da izračunate izgovorite programu sa *Calculate* .... i nakon toga klijent šalje odgovor koji naš program izgovara.

def process\_text(input):

try:

# if ’search’ in input or ’play’ in input:

# search\_web(input.lower())

# return if "who are you" in input or "define yourself" in input: odgovor = ’’’Hello, I am Katarina. Your personal Assistant. I am here to make your life easier. You can command me to perform various tasks such as calculating sums or opening applications etcetra’’’

speak(odgovor)

return

elif "who made you" in input or "created you" in input:

odgovor = "I have been created by Aleksa, Andrea and Lena."

speak(odgovor)

return

elif "calculate" in input.lower():

# u ovom delui smo uneli moj id za wolpframalpha API za racunanje, ovo samo radi za kada mu se kaze calculate

app\_id = "56RVRT-3T6YLGPQQX"

client = wolframalpha.Client(app\_id)

indx = input.lower().split().index(’calculate’)

query = input.split()[indx + 1:]

res = client.query(’ ’.join(query))

answer = next(res.results).text speak("The answer is " + answer)

return

elif ’open’ in input:

# za otvaranje aplikacija

otvori\_aplikaciju(input.lower())

return

else:

speak("I do not understamd, try again!")

return

# speak("I can search the web for you, Do you want to continue?")

# ans = get\_audio()

# if ’yes’ in str(ans) or ’yeah’ in str(ans):

# search\_web(input)

# else:

# return

except :

speak("I did not quite catch that, please try again!")

# speak("I don’t understand, I can search the web for you, Do you want to continue?")

# ans = get\_audio()

# if ’yes’ in str(ans) or ’yeah’ in str(ans):

# search\_web(input)

Komanda otvori aplikaciju je jako jednostavna. U tom delu koda prekopirali smo celokupnu lokaciju aplikacije koje je moguće otvarati. U ovom trenutku implementirali smo Google Chrome, Microsoft Word, Microsoft Excell i Brave.

def otvori\_aplikaciju(input):

if "chrome" in input:

speak("Google Chrome")

os.startfile(’C:\Program Files (x86)\Google\Chrome\Application\chrome.exe’)

return

elif "word" in input:

speak("Opening Microsoft Word")

os.startfile(’C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Microsoft Office 2013\\Word 2013.lnk’)

return

elif "excel" in input:

speak("Opening Microsoft Excel")

os.startfile(’C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Microsoft Office 2013\\Excel 2013.lnk’)

return

elif "brave" in input:

speak("Opening Brave browser")

os.startfile(’C:\Program Files (x86)\BraveSoftware\Brave-Browser\Application\brave.exe’)

return

else:

speak("Application not available")

return

# **ZAKLJUČAK**

Izlaskom Siri na tržište, 2011. godine, voice assistant doživljava ogromnu ekspanziju . Poslednjih nekoliko godina oni postaju sve popularniji i traženiji.

U ovom seminarskom radu, započeli smo samo početnu verziju naše aplikacije sa osnovnim funkcionalnostima. Po uzoru na mnoge, danas veoma poznate i rasprostranjene aplikacije širom sveta (kao što su Siri, Alexa, Hound…) voleli bismo da i mi jednog dana lansiramo Katarinu koja će do tada obuhvatati sve najnovije mogućnosti i koja će redovno pratiti trendove. Prvi korak u našem daljem razvoju biće obezbedjivanje “surfovanja internetom”. Nakon toga bismo uvodili mapu koja bi mogla da nas usmerava na najkraći put do željene destinacije. To su neke od početnih ideja, a mogućnosti daljeg razvoja su ogromne.

Smatramo da je ova tema veoma interesantna i perspektivna i da će ovakve stvari za par godina biti deo svakodnevnice svakog od nas.

# **KORIŠĆENA I REFERENTNA LITERATURA**

1. Python pyaudio.PyAudio() Examples.[Online].Available:<https://www.programcreek.com/python/example/52624/pyaudio.PyAudio?fbclid=IwAR1GHt08PegKQC6S4JrHDQpelwWfBcXXCj_Z3uKUle0grrohUdZefG50QUM>
2. How to get started with Google Text-to-Speech using Python.[Online].Available: <https://towardsdatascience.com/how-to-get-started-with-google-text-to-speech-using-python-485e43d1d544>
3. Python | Create a simple assistant using Wolfram Alpha API.[Online].Available: <https://www.geeksforgeeks.org/python-create-a-simple-assistant-using-wolfram-alpha-api/?fbclid=IwAR2C3bcM9BEf46iXi6C3fcW7_XTStUWeTJSvEBd8KqS2WdfB0K-xasCnpCE>