

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Primož Pesjak

**Mobilna aplikacija za najemanje vozil  
z uporabo asistenta**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM  
PRVE STOPNJE  
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Rok Rupnik

Ljubljana, 2019

COPYRIGHT. Rezultati diplomske naloge so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavo in koriščenje rezultatov diplomske naloge je potrebno pisno privoljenje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

*Besedilo je oblikovano z urejevalnikom besedil L<sup>A</sup>T<sub>E</sub>X.*

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Proučite nekaj asistentov, ki jih je kot storitev možno uporabiti v mobilni aplikaciji. Ko izberete najbolj primernega asistenta, zasnujte mobilno aplikacijo, ki bo omogočala najem vozil. Pri tem uporabite že obstoječo platformo za najem vozil: mobilna aplikacija in asistent naj uporabljata storitve že obstoječe platforme za najem vozil.



*Zahvaljujem se mami, očetu in vsem babicam in dedkom, ki so mi omogočili študij. Matej Grošelj, Živa Lampret, Mitja Stovanje, Žiga Strgar, Matej Hacin, Sebastijan Mislej ter Rok Rupnik - za ogromno podporo in za mentorstvo pri pisanju diplomske naloge. Zahvala gre tudi podjetju Avant car, ki mi je omogočilo uresničitev projekta.*



# Kazalo

Povzetek

Abstract

<b>1</b>	<b>Uvod</b>	<b>1</b>
1.1	Poslovna domena . . . . .	1
1.2	Struktura diplomske naloge . . . . .	1
<b>2</b>	<b>Pregled problema in rešitve</b>	<b>3</b>
2.1	Problem . . . . .	3
2.2	Rešitev . . . . .	3
2.3	Analiza trga . . . . .	4
<b>3</b>	<b>Tehnologije in programska oprema</b>	<b>5</b>
3.1	Web storm . . . . .	5
3.2	Izvršitev z Node.js knjižnico . . . . .	5
3.3	Axios . . . . .	5
3.4	SSML . . . . .	6
3.5	Firebase in Cloud Functions . . . . .	7
3.6	Predstavitev platforme DialogFlow-a . . . . .	9
3.7	Spletna stran Actions on Google . . . . .	15
3.8	Ngrok . . . . .	17
<b>4</b>	<b>Načrtovanje in razvoj pogovornega robota</b>	<b>19</b>
4.1	Načrt . . . . .	19

4.2	Odločitev za izbor Googlovega asistenta . . . . .	21
4.3	Logika delovanja sistema . . . . .	22
4.4	Opis poteka razvoja rešitve . . . . .	23
4.5	Delovanje Dialogflow-a . . . . .	24
4.6	Uporabljeni podatki iz API-ja . . . . .	27
4.7	Glasovni pogovorni robot . . . . .	28
4.8	Glasovni uporabniški vmesnik . . . . .	29
4.9	Učinkovita komunikacija . . . . .	29
4.10	Prepoznavanje govora in glasovni pomen . . . . .	30
<b>5</b>	<b>Ključne funkcionalnosti s primeri srečne poti</b>	<b>31</b>
5.1	Neposredna rezervacija . . . . .	32
5.2	Izbira vozila, glede na lokacijo . . . . .	33
5.3	Upravljanje aktivne rezervacije . . . . .	36
5.4	Obvladovanje napak med pogovorom . . . . .	37
<b>6</b>	<b>Možnosti nadaljnega razvoja in monetizacija</b>	<b>39</b>
6.1	Nadaljnji razvoj . . . . .	39
6.2	Možnosti monetizacije . . . . .	41
<b>7</b>	<b>Zaključek</b>	<b>43</b>
	<b>Literatura</b>	<b>45</b>

# Seznam uporabljenih kratic

kratica	angleško	slovensko
<b>AI</b>	Artificial Intelligence	Umetna inteligenca
<b>RFID</b>	Radio Frequency IDentification	Radiofrekvenčna identifikacija
<b>IDE</b>	Integrated Development Environment	Razvojno kolje
<b>SSML</b>	Speech Synthesis Markup Language	Označevalni jezik za sintetični govor
<b>TTS</b>	Text to Speech	Besedilo v govor
<b>ML</b>	Machine learning	Strojno učenje
<b>NLP</b>	Natural language processing	Obdelava naravnega jezika
<b>ASR</b>	Automatic speech recognition	samodejno razpoznavanje govora
<b>AI</b>	Artificial Intelligence	Umetna inteligenca
<b>API</b>	Application Programming Interface	Vmesnik za programiranje
<b>JSON</b>	JavaScript Object Notation	Javascript objektni zapis
<b>XML</b>	Extensible Markup Language	Razširljivi označevalni jezik
<b>URL</b>	Uniform Resource Locator	Enolični krajevnik vira
<b>UI</b>	User Interface	Uporabniški vmesnik
<b>NLU</b>	Natural Language Understanding	Razumevanje naravnega jezika
<b>VUI</b>	Voice User Interface	Glasovni uporabniški vmesnik



# Povzetek

**Naslov:** Mobilna aplikacija za najemanje vozil z uporabo asistenta

**Avtor:** Primož Pesjak

Diplomsko delo obravnava področje samopostrežne kratkoročne izposoje električnih vozil (t. i. souporaba vozil), ki je možna 24 ur dnevno vse dni v tednu s pomočjo aplikacije na pametnem telefonu. Cilj diplomskega dela je razviti pogovornega robota (angl. chatbot) za glasovno komunikacijo, s čimer bomo poenostavili in pospešili proces najema vozila. Uporabniki bodo z njegovo pomočjo lahko prostoročno rezervirali vozilo. Uporabili bomo naj-novejšo tehnologijo za razvoj pogovornega robota, ki bo na voljo v okviru Google asistenta, z namenom čimvečje dostopnosti.

**Ključne besede:** pogovorni robot, souporaba vozil, asistent, promet, glasovna rešitev.



# Abstract

**Title:** Mobile application for hiring vehicles using an assistant

**Author:** Primož Pesjak

This thesis deals with the field of self-service short-term rental of electric vehicles (i.e. vehicle sharing) which is possible 24 hours a day, every day of the week, with the help of a smart phone application. The goal of the thesis is to develop a chatbot for a voice communication, which will simplify and speed up the process of renting a vehicle. Users will be able to reserve a vehicle free of charge with chatbot's help. We will use the latest development technology for the development of the chatbot, which will be available through the Google Assistant to increase the accessibility.

**Keywords:** chatbot, car sharing, assistant, traffic, voice solution.



# Poglavje 1

## Uvod

### 1.1 Poslovna domena

Diplomsko delo obravnava področje samopostrežne kratkoročne izposoje električnih vozil (t. i. souporaba vozil), ki je možna 24 ur dnevno vse dni v tednu s pomočjo aplikacije na pametnem telefonu. Uporabnik v aplikaciji na zemljevidu vidi razpoložljiva vozila, si izbere prosto vozilo na želeni lokaciji, ga rezervira, aktivira in po koncu najema uporabo tudi plača. Souporaba vozil v svetu predstavlja enega od najhitreje rastočih segmentov avtomobilske industrije, saj je del delitvene ekonomije, kjer ob pomoči tehnologije jasno razmejiš uporabnost vozila tudi na časovna področja, krajša od enega dneva, in zaračunaš plačilo po dejanski porabi, kar lahko za uporabnika pomeni nižje stroške za mobilnost.

V diplomskem delu smo izhajali iz vprašanja, ali nam lahko tehnologija pomaga k še hitrejšemu in enostavnejšemu dostopu do najema vozil iz souporabe [6].

### 1.2 Struktura diplomske naloge

Diplomsko delo pričenjamo s predstavitvijo poslovne domene, nato opredelimo raziskovalni problem in opišemo možno rešitev zanj. V našem pri-

meru gre za poenostavitev in pospešitev procesa najema vozila prek aplikacije z uporabo glasu. V zvezi s tem najprej analiziramo trenutni trg za t. i. pogovornega robota (angl. chatbot). V naslednjem poglavju opišemo najpomembnejše tehnologije, uporabljene za izvršitev projekta, in ključne funkcionalnosti pogovornega robota. Sledi poglavje z opisom platforme Dialogflow ter predstavitev delovanja in povezanosti posameznih komponent. Zadnji vsebinski del namenimo priložnostim za nadaljnji razvoj in monetizacijo rešitve.

V nadaljevanju opišemo načrt celotnega sistema, njegovo povezavo s pogovornim robotom, vrsto uporabljenih podatkov in odločitev za ustreznega pogovornega robota.

# Poglavje 2

## Pregled problema in rešitve

### 2.1 Problem

Ker so sistemi delitvene ekonomije (med njimi tudi souporaba vozil) v hitrem porastu, postaja vprašanje intuitivne in preproste uporabniške izkušnje čedalje pomembnejše. Doslej so se uveljavili sistemi, ki delujejo predvsem preko RFID kartic in aplikacij, pri čemer pričujoče diplomsko delo izhaja iz možnosti najema vozila preko aplikacije, ki meri porabljeni čas ter prevožene kilometre in nato na tej osnovi zaračuna končni strošek mobilnosti. Izhajajoč iz doslej navedenega opredeljujem problem diplomskega dela in sicer: kako lahko proces kratkoročnega samopostrežnega najema vozila, ki poteka preko aplikacije na pametnem telefonu naredimo uporabniku še hitrejšega in preprostejšega.

### 2.2 Rešitev

Glede na v prejšnjem podpoglavju opredeljeni problem predlagamo in v nadaljevanju dela tudi proučimo naslednjo rešitev: aplikacijo za Google asistenta, s pomočjo katere lahko rezerviramo vozilo z uporabo glasu ali prek kratkih tekstovnih zapisov. Razlog vidimo v tem, da je tehnologija že tako napredovala, da zagotavlja dovoljšno natančnost – v angleškem jeziku je tovrstnih napak

manj kot 3%, pri čemer po statistiki QuoraCreative opravimo že 25% iskanj s pomočjo glasu na mobilnih napravah. Izhajamo iz predpostavke rastoče uporabe glasu za komunikacijo z napravami – odstotek se bo še povečeval, saj nam je pogovor bližji, bolj naraven ter hitrejši kot »mehansko« pisanje. S to rešitvijo uporabnikom omogočimo, da lahko rezervirajo vozilo s pomočjo glasu, kar predstavlja hitrejšo, bolj intuitivno in tudi preprostejšo možnost. [26, 11].

## 2.3 Analiza trga

Tržišče govornega upravljanja se iz dneva v dan povečuje, saj nam asistenti, kot so Google Assistant, Siri in Alexa, vse bolj olajšujejo življenje in postajajo zmeraj bolj integrirani vanj. Ne glede na to, ali je to v hiši za vklop naprav za zabavo, iskanje informacij, ki nas zanimajo, ali pa opravljajo rezervacije za frizerja, kot je predstavil Google na svoji konferenci Google I/O 2017. Zanimanje za te naprave in glasovne asistente se povečuje, kar je potrdila tudi raziskava podjetja Juniper Research. Odkrili so, da se bo njihova priljubljenost v prihodnje kar za nekajkrat povečala. S trenutnih 25 milijonov uporabnikov naj bi njihovo število do leta 2023 naraslo na 275 milijonov, kar je več kot 1.000% rast. Po njihovem mnenju je trenutno najboljši asistent Alexa. Njegova največja trenutna prednost je preprostost ustvarjenja t. i. skillov, ki so raznorazne aplikacije. Teh je preko 50.000, medtem ko jih ima Google Asistent precej manj.

Jezik asistentov je zaenkrat še primarno angleški, z leti, ko bodo uradno podprli še Slovenijo, pa bo zadeva postala priljubljena tudi pri nas. To pomeni, da bomo s pomočjo glasu lahko rezervirali vozilo, si naročili večerjo ali pa taksi tudi v slovenskem jeziku. Uporaba pogovornih robotov je brezplačna, edine omejitve pri odločitvi za njegovo uporabo so: regija, jezik in morebitna prijava v pogovornega robota. Aplikacija za izposajo vozil s pomočjo glasovnih ukazov bi tako bila med prvimi tovrstnimi aplikacijami v Sloveniji [15, 1].

## Poglavje 3

# Tehnologije in programska oprema

### 3.1 Web storm

Webstorm je razvojno okolje za delo z JavaScriptom, ki ga je razvilo podjetje IntelliJ. Programsko kodo bi lahko sicer pisali tudi v Atomu, Sublime ali pa celo v beležnici, vendar je delo enostavnejše, če je v ozadju kakšen IDE, ki skrbi za formatiranje kode in urejanje projekta [28].

### 3.2 Izvršitev z Node.js knjižnico

Gre za priporočeno knjižnico Googla za dostopanje in interakcijo s platformo Actions on Google, če ustvarimo izvršitev webhook v JavaScriptu. Z njeno pomočjo lahko lažje namestimo izvršitev webhook, ker knjižnica poskrbi za nizko nivojsko komunikacijo s platformo Actions on Google [20].

### 3.3 Axios

Axios je knjižnica, ki temelji na Promisu za NodeJS. Promise predstavlja vrednost, ki ni nujno znana, ko je ustvarjena. To omogoča, da je lahko

asinhrona in lahko potem povežemo Handlerje z njo če je uspešna, končana, ali pa zavrnjena. S tem dosežemo, da namesto da takoj vrne končno vrednost, asinhrona metoda vrne Promise, da bo v prihodnosti dobila vrednost, zato Google Asistent čaka, da dobi odziv. V projektu se uporablja, za klicanje na naš API, kjer mora asistent počakati, da dobi rezultat nazaj preden je na vrsti spet uporabnik [22, 2, 21].

```
let clientInitGetNewLocations = function (conv) {
  storage.resetCurrentCar(conv);
  return client(conv).get( url: 'locations?populate=%5B%22companyID%22%5D&fields=["_id", "name", "distance", "geoLocation"]')
    .then( onfulfilled: (response) => {
      let userLatitude = storage.getUserLatitude(conv);
      let userLongitude = storage.getUserLongitude(conv);
      let carLocations = fun.getAvailableLocationsDistanceSorted(response.data, userLatitude, userLongitude);

      storage.setLocations(conv, carLocations);
      storage.setLocationIndex(conv, newIndex: 0);
      storage.setCurrentLocation(conv, storage.getLocations(conv)[0]);
      return clientGetLocationsDetails(conv);
    }).catch( onrejected: function (error) {
      return conv.close(`${error.message}`);
    });
};
```

Slika 3.1: Primer Promise kode za pridobitev vseh lokacij z vozili na API

## 3.4 SSML

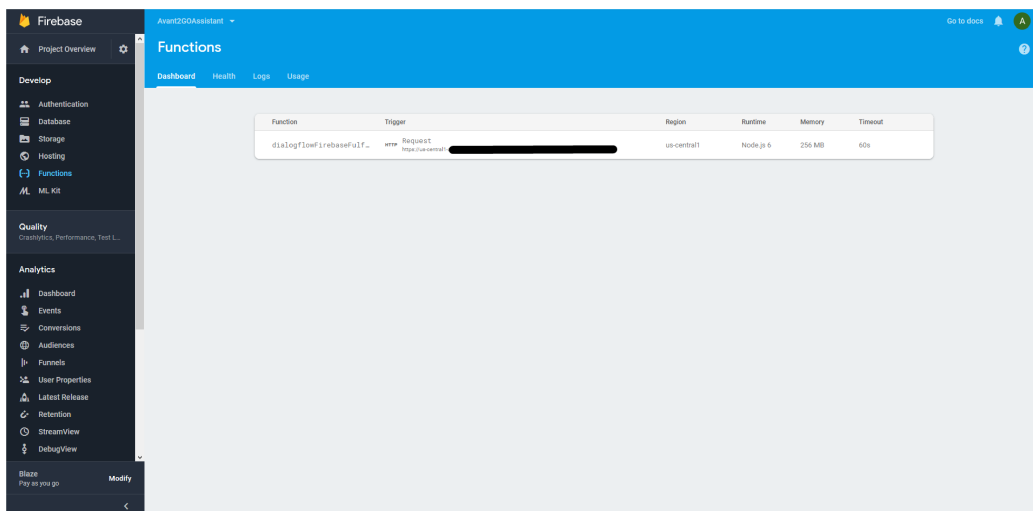
S pomočjo SSML-ja dosežemo večji nadzor nad govorom. SSML je markup jezik. Dialogflow že ima integrirane svoje privzete načine govora, npr. da počaka v primeru pike ali vejice, vendar to ni vedno dovolj. SSML nam omogoča tudi, da predvajamo zvoke, vstavljamo prekinitve v govor, višino, globino, moč, črkovanje ipd. Pogovornemu robotu smo povečali hitrost govora za 10%, ker s tem dosežemo bolj naraven govor. SSML lahko testiramo v simulatorju TTS (angl. text to speech), ki se nahaja v konzoli Actions on Google pod zavihkom »Zvok« [27, 25].

```
< speak >
  Here are < say-as interpret-as="characters">SSML</ say-as > samples.
  I can pause < break time="3s" />.
  I can play a sound
  < audio src="https://www.example.com/MY_MP3_FILE.mp3">didn't get your MP3 audio file</ audio >.
  I can speak in cardinals. Your number is < say-as interpret-as="cardinal">10</ say-as >.
  Or I can speak in ordinals. You are < say-as interpret-as="ordinal">10</ say-as > in line.
  Or I can even speak in digits. The digits for ten are < say-as interpret-as="characters">10</ say-as >.
  I can also substitute phrases, like the < sub alias="World Wide Web Consortium">W3C</ sub >.
  Finally, I can speak a paragraph with two sentences.
  < p >< s >This is sentence one.</ s >< s >This is sentence two.</ s ></ p >
</ speak >
```

Slika 3.2: Primer SSML-ja

### 3.5 Firebase in Cloud Functions

Firebase je platforma, kamor se naložijo aplikacije, brez potrebnega znanja programiranja za strežnike, zato je razvoj hitrejši in lažji za nove uporabnike. Podprte platforme so: splet, iOS, Android in druge. Firebase Cloud Functions samodejno požene kodo glede na to, kateri dogodek se sproži. Koda se objavi v Firebasov oblak, zato ni potrebe, da imamo svoj strežnik za aplikacijo. Videz nadzorne plošče prikazuje slika 3.5. Na spletnem mestu, ki je podan pod Request URL-naslovom, je datoteka index.js, ki je zaradi varnosti skrita. Konzola se uporabi za beleženje podatkov, preverjanje uporabe in delovanja funkcije, ki smo jo objavili.



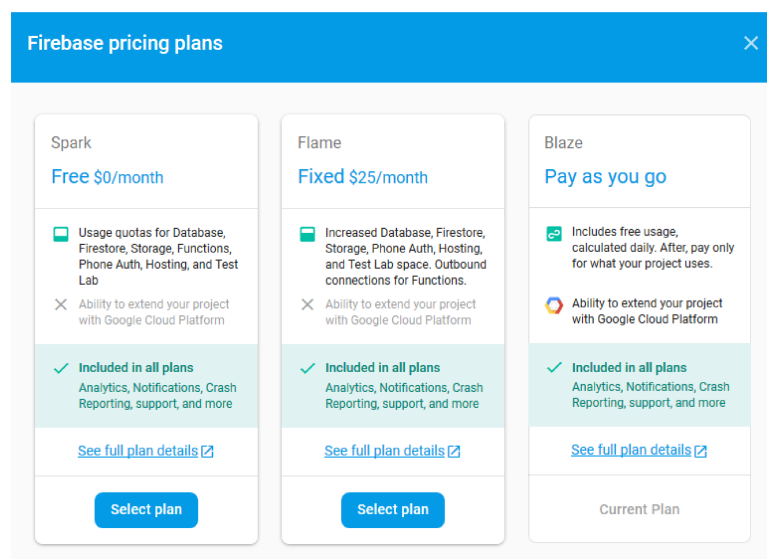
Slika 3.3: Firebase Cloud Functions nadzorna plošča

Firebase nam ponuja več različnih načinov uporabe in vsak ima svojo ceno [9].

**Brezplačna** različica je omejena predvsem na API-strani. Ne moremo opravljati kakršnihkoli klicev na naš API, ampak samo na Googlovega, kar je zelo omejujoče.

**Flame** je plačljiv in ima odklenjene vse funkcionalnosti, razen klicev na strežnik.

**Blaze** je prav tako plačljiv, vendar plačaš po porabi in omogoča klice na lasten API. Ta različica je v času razvoja brezplačna, zato smo jo uporabili tudi za naš projekt.



Slika 3.4: Firebase-ove cenovne možnosti

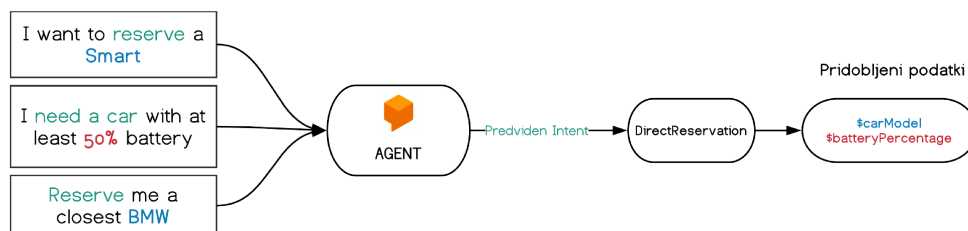
## 3.6 Predstavitev platforme DialogFlow-a

Najpomembnejša tehnologija od vseh uporabljenih je Dialogflow. Gre za spletni vmesnik, kjer lahko gradimo in testiramo pogovore. Dialogflow omogoča prepoznavo glasu, razumevanje naravnega jezika ter pretvorbo besedila v govor.

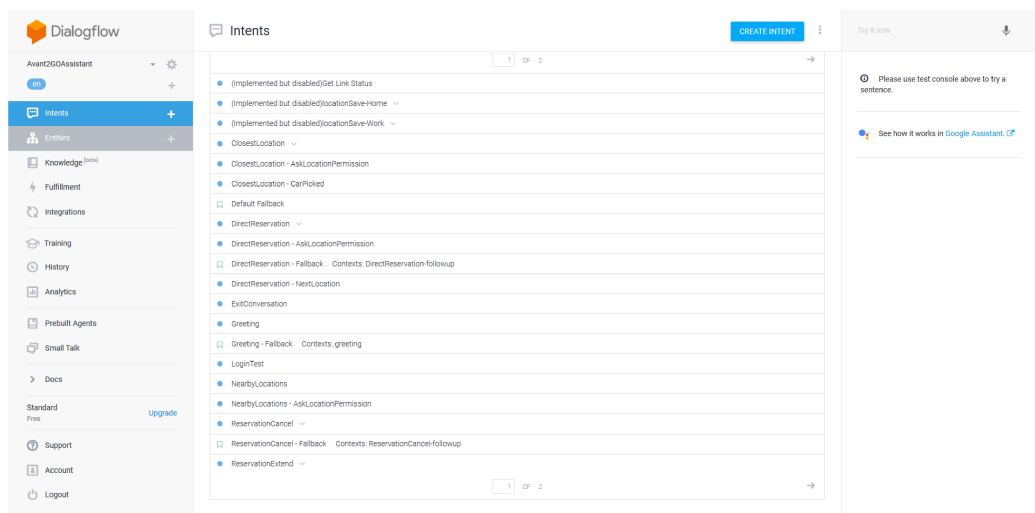
Na začetku se je imenoval API.AI, njegov cilj pa je bil, da lahko razvijalci s pomočjo API-ja dodajo NLP (angl. natural language processing) na naprave, aplikacije in storitve. Konec leta 2017 se je API.AI preimenoval v Dialogflow in še vedno ostaja platforma za razvijanje interakcij med človekom in računalnikom, ki uporablja NLP in strojno učenje. Agenti bi lahko najlažje opisali kot modul NLU. Vse agente, ki jih zgradimo s pomočjo Dialogflowa, lahko integriramo v številne pametne naprave oz. velja, kjer je naložen Google Asistent, tam je lahko naš agent. [7, 8]

Agent je drugo ime za model NLU, ki je lahko vključen v aplikacijo, spletno stran, produkt ali storitev. Ta prevede vhodne podatke, ki jih je prejel od uporabnika, v uporabne podatke. To se zgodi takrat, ko najde ujemanje s kakšnim ustvarjenim »intentom« v tem agentu, kot je prikazano na sliki 3.6.

Agentu definiramo vse entitete, intente, kontekste in drugo. Najpomembnejši so intenti, ker z njimi mapiramo uporabniške stavke in uporabniku lahko s tem zagotovimo pravi odziv [9].



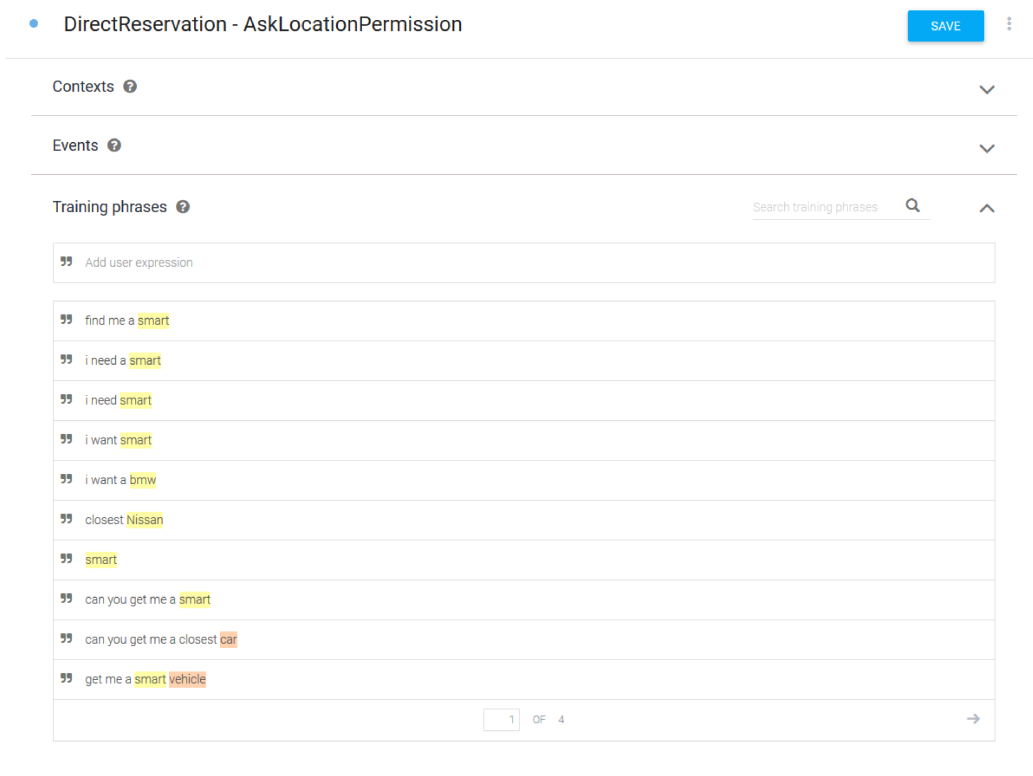
Slika 3.5: Predstavitev delovanja agenta



Slika 3.6: Agent v Dialogflowu z intenti

Agent je sestavljen iz več komponent:

**Intent**, ko uporabnik postavi vprašanje ali se kakorkoli odzove, bo Dialogflow poskušal najti pravega, ki ima pravi odgovor na to, kar uporabnik hoče od njega. Zato mu moramo dati čim več učnih primerov, kako pride do njega. Več kot jih je, boljše bo agent razumel uporabnika.



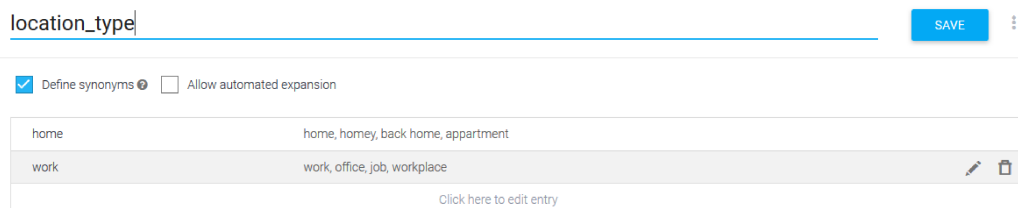
The screenshot displays the Dialogflow console interface for an intent named "DirectReservation - AskLocationPermission". At the top right, there is a blue "SAVE" button and a vertical ellipsis menu. Below the intent name, there are sections for "Contexts", "Events", and "Training phrases". The "Training phrases" section is expanded, showing a list of phrases with highlighted entities. The phrases are:

- find me a smart
- i need a smart
- i need smart
- i want smart
- i want a bmw
- closest Nissan
- smart
- can you get me a smart
- can you get me a closest car
- get me a smart vehicle

At the bottom of the list, there is a pagination indicator showing "1 OF 4" and a right-pointing arrow.

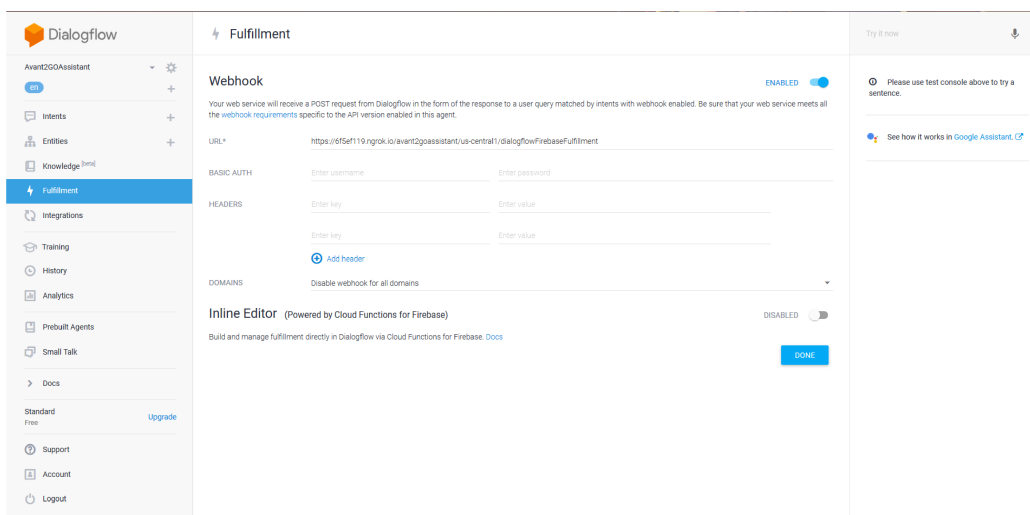
Slika 3.7: Primer Intenta

**Entiteta** so spremenljivke, ki jih lahko sami ustvarimo. V našem primeru smo ustvarili entiteto »carmodels«, ki je vključevala avtomobile Renault, Zoe, Smart in BMW. Kadarkoli je uporabnik izrekel to ime, je bila prepoznana ta entiteta tipa »carmodels«.



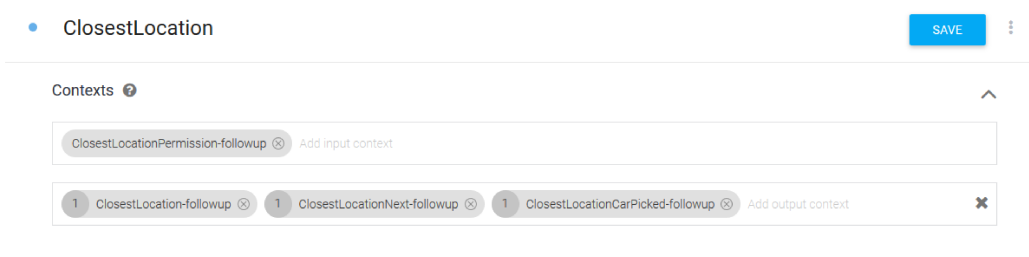
Slika 3.8: Primer entitete

**Izvršitev** deluje prek Google Cloud Functions. To je rešitev, kamor lahko damo kodo, da se izvaja, ne da bi mi skrbeli za strežnike. Ko moramo dodati svojo kodo, npr. spletne zahteve našega API-ja, to storimo v kodi, ki se neodvisno izvaja na strežnikih.



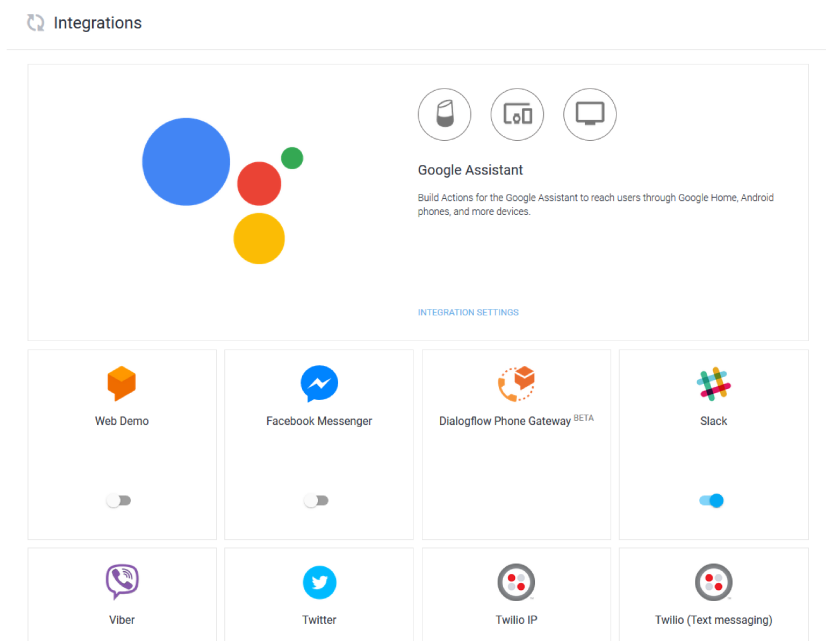
Slika 3.9: Primer Izvršitve

**Kontekst**, z njim lahko vodimo pogovore in držimo uporabnika na pravi, t. i. srečni poti (angl. happy path). Uporabljamo ga torej za to, da agent ve, o čem se pogovarjamo, in ne daje drugih odgovorov kot na trenutno vprašanje. Če trenutno rezerviramo avto in nas vpraša, ali hočemo avto, in mi rečemo, da ga hočemo, on ve, da hočemo ta avtomobil in ne katerega drugega. Če bi agent bil brez konteksta, ne bi vedel, kaj je mislil z »da«, in ne bi odgovoril na tak način, kot hočemo.



Slika 3.10: Primer kontekstov

**Integracija platforme** - možna je integracija z več kot 20 platformami, kot so Messenger, katerakoli spletna stran, Slack ipd., če to želimo. Najbolj uporabljena pa je integracija z Googlovim asistentom.



Slika 3.11: Zavihek integracija

Dialogflow je večjezičen, kar pomeni, da agentu lahko določimo jezik glede na državo ali regijo oz. glede na to, kar ima uporabnik nastavljeno. Na začetku imamo en primarni jezik, nato pa jih lahko dodajamo. Največje število jezikov na agenta je 15. Trenutno podpira 19 jezikov, če izvzamemo tiste, ki so ločeni glede na regijo. To so: kitajščina, danščina, nizozemščina, angleščina, francoščina, nemščina, hindijščina, indonezijsščina, italijanščina, japonsščina, korejščina, norveščina, portugalsščina, ruščina, španščina, švedščina, tajščina, turščina in ukrajinsščina.

Slovenskega jezika trenutno še ni med njimi. Napovedali so, da bodo v naslednjem letu integrirali še 30 drugih jezikov [16, 17].

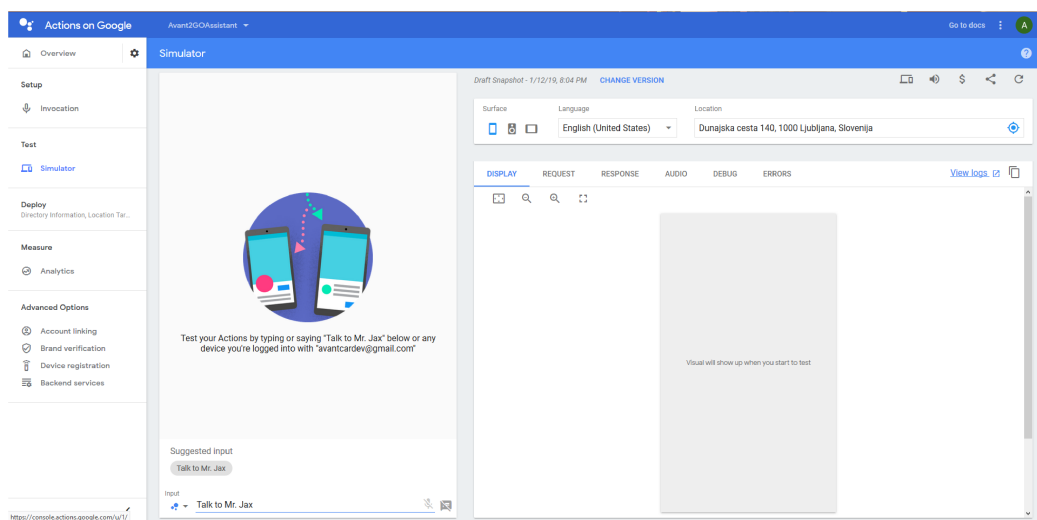
### 3.7 Spletna stran Actions on Google

Actions on Google je platforma, ki dovoljuje objavljane programske opreme, t. i. Actions. Gre za aplikacije, ki razširjujejo Googlovega asistenta. Te aplikacije uporabniki vidijo. Prej smo omenili agenta; če želimo, da agent postane viden, ga moramo objaviti na Actions on Google, kjer postane t. i. Action; to je na koncu ta aplikacija, ki jo najdejo uporabniki. Obstajata dva načina, da uporabniki najdejo Action:

**Neposredno**, ta način pride v poštev, ko uporabnik pozna ime aplikacije in jo pokliče po imenu. Takrat poskuša Google asistent matchat z ustrežno Action in če jo najde, se aplikacije odpre, v nasprotnem primeru pa postane to navadno iskanje na Googlu. Primer: »Ok Google, Talk to Car sharing assistant.«

**Posredno**, kadar uporabnik pokliče eno funkcionalnost aplikacije. Google asistent najde ustrezen Action, ki bi lahko ustrezala tej zahtevi, in se neposredno v Actionu pokliče, kot da bi se že pogovarjali s tem agentom. V tem primeru se preskoči nastavljena dobrodošlica in se neposredno sproži ustrezen intent. Primer: »Ok Google, tell Car sharing assitant that I need a Smart.«

Spletna stran Actions on Google nam omogoča tudi simuliranje pogovora na spletu. Prek te spletne strani lahko objavimo aplikacijo javnosti in določimo, na kakšen način bodo uporabniki našli Action. Ne pokažejo se napake, ki se mogoče lahko zgodijo, zato se za razvoj pogovornega robota uporabi še Ngrok.



Slika 3.12: Actions on Google spletna stran

### 3.8 Ngrok

Ngrok ustvari javno dostopen URL-naslov do lokalnega strežnika (strežnika, ki teče na našem računalniku). Uporabljali smo ga za testiranje webhookov, da smo lahko lokalno testirali. Ob zagonu terminala Ngrok in z ukazom »ngrok http 80« se vzpostavi predor povezava do lokalnega strežnika. Sedaj je dostopen prek javnega naslova, ki je v obliki: »https://92832de0.ngrok.io«. Ta URL prekopiramo v Dialogflow, da ne bo več klical Firebase, ampak lokalni strežnik. Največji prednosti sta lokalno testiranje in večji nadzor pretoka podatkov med testnim uporabnikom in razvito kodo. Takšna oblika testiranja je prinesla instantne rezultate, medtem ko je z Google Functions za pričakovani rezultat treba počakati tudi do 3 minute [19].

```

ngrok by @tjones/reveable (Ctrl+C to quit)
ngrok status
Session Expires 7 hours, 57 minutes
Version 2.2.4
Region United States (us)
Web Interface https://228.0.0.1:4040
Forwarding http://4ef0e1.ngrok.io -> localhost:5000
Connections
t1l  opn  r1t1  r1t5  p1s0  p1s0
-----
5      0      0.35  0.42  0.14  0.54
HTTP Requests
-----
POST /avant2goassistant/us-central1/dialogflow/firebaseFullfillment 200 OK
POST /avant2goassistant/us-central1/dialogflow/firebaseFullfillment 200 OK
POST /avant2goassistant/us-central1/dialogflow/firebaseFullfillment 200 OK
POST /avant2goassistant/us-central1/dialogflow/firebaseFullfillment 200 OK
POST /avant2goassistant/us-central1/dialogflow/firebaseFullfillment 200 OK

{"coordinates": {"latitude": 41.9044915, "longitude": 14.668567799999999}, "id": "ABpppE1Fdw7G0uCd4p7xQ07YX7AH83p8k2_0XX8ef_L5-uYcPqH91nL1pJ9mH1G1yA2Bef7x10DU", "type": "dialogflow", "screen": true, "body": {"discarded": true, "version": 2, "action": "NearbyLocations:Handler", "intent": "NearbyLocations", "parameters": {}, "context": {"session": "projects/avant2goassistant/agent/sessions/ABpppE1Fdw7G0uCd4p7xQ07YX7AH83p8k2_0XX8ef_L5-uYcPqH91nL1pJ9mH1G1yA2Bef7x10DU", "input": {}}, "output": {}}, "incoming": {"query": {"text": "actions_intent_PERMISSION", "data": {}}, "query": "actions_intent_PERMISSION", "data": {}}, "saved locations": [{"status": "end", "headers": {"content-type": "application/json;charset=utf-8"}, "body": {"payload": {"suggestions": [{"expectUserResponse": true, "iriResponse": {"items": [{"simpleResponse": {"textToSpeech": "Speak", "voiceGender": "male", "variant": "2"}, "closest locations are (id) - (1)1544 cesta 12.741 kilometers, Poslovalni center Nova Polje 12.795 kilometers, Jurean - gostilna in pizzeria 13 kilometers. Anything else?/prosody>in /voice>in /speak"}], "suggestions": [{"title": "Closest car"}, {"title": "Cars at closest location"}]}]}}, "userStorage": [{"data": [{"currentReservationID": ""}], "carModel": [{"Renault": [{"id": "58d8a9c9138ac01c11227", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11228", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11229", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11230", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11231", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11232", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11233", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11234", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11235", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11236", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11237", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11238", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11239", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11240", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11241", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11242", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11243", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11244", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11245", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11246", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11247", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11248", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11249", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11250", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11251", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11252", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11253", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11254", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11255", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11256", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11257", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11258", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11259", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11260", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11261", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11262", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11263", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11264", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11265", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11266", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11267", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11268", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11269", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11270", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11271", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11272", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11273", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11274", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11275", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11276", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11277", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11278", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11279", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11280", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11281", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11282", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11283", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11284", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11285", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11286", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11287", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11288", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11289", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11290", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11291", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11292", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11293", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11294", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11295", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11296", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11297", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11298", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11299", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11300", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11301", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11302", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11303", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11304", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11305", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11306", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11307", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11308", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11309", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11310", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11311", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11312", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11313", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11314", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11315", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11316", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11317", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11318", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11319", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11320", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11321", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11322", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11323", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11324", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11325", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11326", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11327", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11328", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11329", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11330", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11331", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11332", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11333", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11334", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11335", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11336", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11337", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11338", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11339", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11340", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11341", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11342", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11343", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11344", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11345", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11346", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11347", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11348", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11349", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11350", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11351", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11352", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11353", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11354", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11355", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11356", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11357", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11358", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11359", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11360", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11361", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11362", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11363", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11364", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11365", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11366", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11367", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11368", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11369", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11370", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11371", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11372", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11373", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11374", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11375", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11376", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11377", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11378", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11379", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11380", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11381", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11382", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11383", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11384", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11385", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11386", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11387", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11388", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11389", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11390", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11391", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11392", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11393", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11394", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11395", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11396", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11397", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11398", "id": "58dbafcc6b72f667837e1be"}]}, {"id": "58d8a9c9138ac01c11399", "id": "58dbafcc6b72f667837e1be"}, {"id": "58d8a9c9138ac01c11400", "id": "58dbafcc6b72f667837e1be"}]}

```

Slika 3.13: Izgled ngrok konzole, ter spremljanje zahtev v standardni konzoli



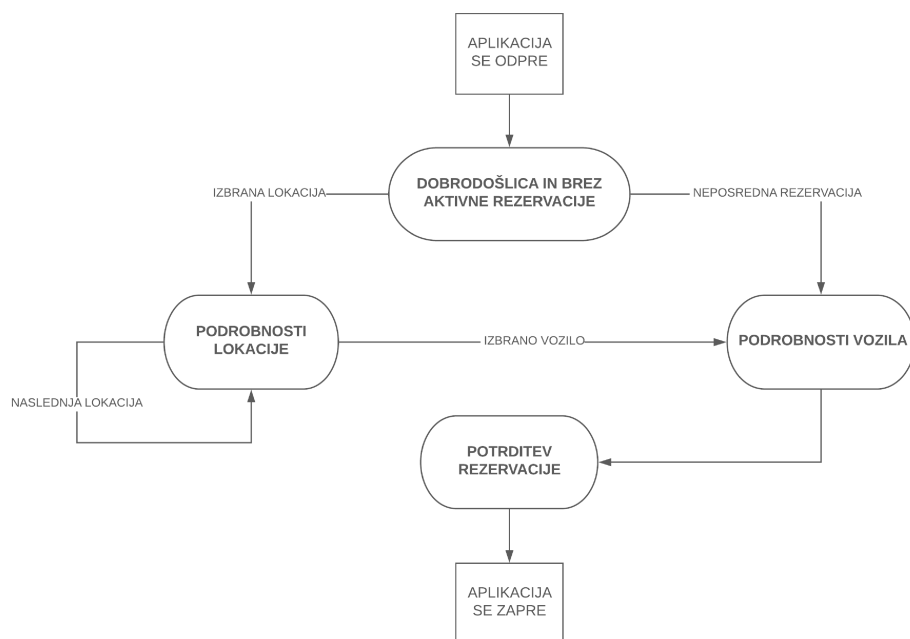
## Poglavje 4

# Načrtovanje in razvoj pogovornega robota

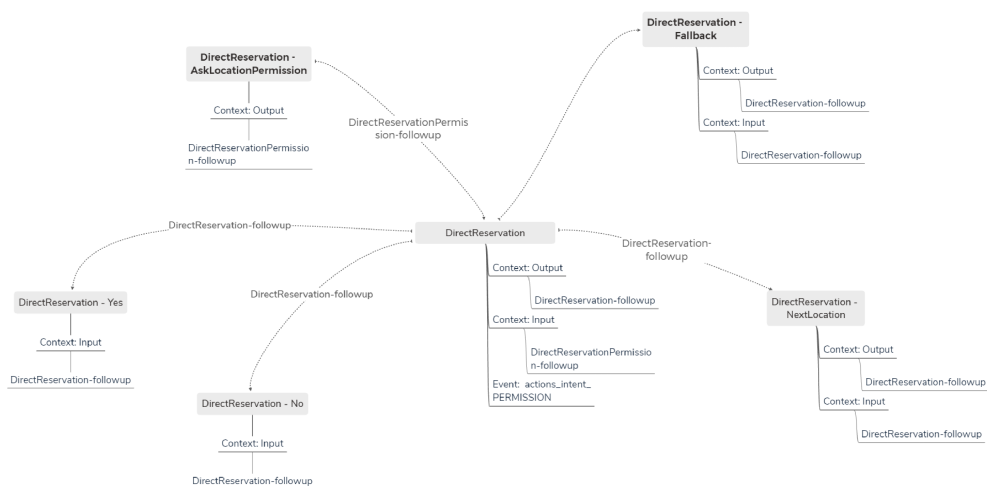
### 4.1 Načrt

Na začetku smo definirali primer uporabe pogovornega robota. Za osnovne potrebe bi aplikacija uporabnika na začetku pozdravila in se predstavila, nato pa ga s posameznimi koraki vodila do uspešne rezervacije vozila. Proučili smo razpoložljive tehnologije, da se ta primer uporabe izvrši, in naleteli na omenjeni Dialogflow, ki je namenjen za ustvarjanje pogovornih robotov za Google Asistenta.

Zaradi lažje predstavitve kompleksnosti aplikacije je bilo treba ustvariti diagrame za posamezne intente, kajti Dialogflow ponuja samo tabelarno predstavitev. Na sliki 4.1 je intent neposredne rezervacije.



Slika 4.1: Diagram, ki prikazuje, kako bi lahko uporabnik komuniciral s pogovornim robotom



Slika 4.2: Prikaz direktne rezervacije z intenti

## 4.2 Odločitev za izbor Googlovega asistenta

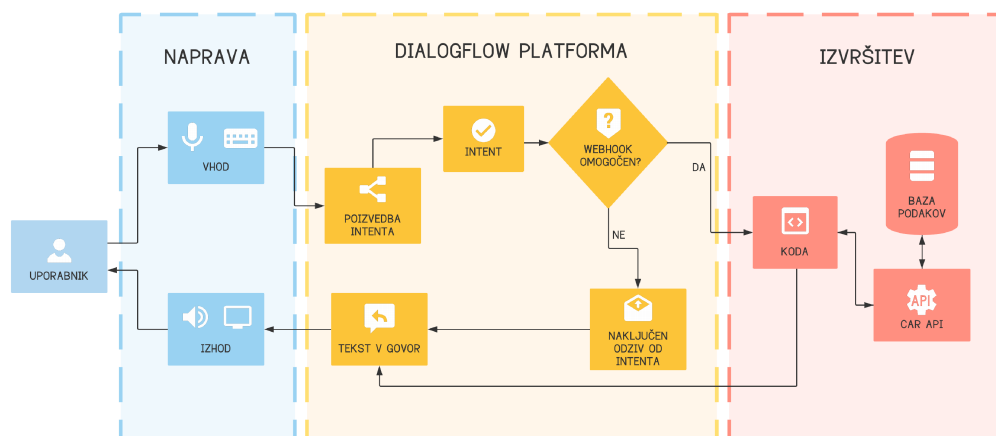
Gre za glasovno upravljanega Googlovega pametnega asistenta, ki obstaja od maja 2017. Prvotno je bil mišljen kot razširitev Google Now, kjer gre za kartice, ki te obveščajo o novicah okoli tebe, odvisno od tega, kar te zanima. Googlov asistent omogoča naslednje funkcionalnosti:

- posredovanje informacije o vremenu
- posredovanje najnovejših novic
- zagon štoparice
- glasovno igranje igrice
- naročanje hrane
- predvajanje glasbe
- vklop televizorja ali ugašanje luči
- naročanje prevoza do doma itd.

Komuniciramo lahko pisno ali govorno. Največja prednost je ravno to socializiranje, saj je Google Asistent najbolj blizu človeškega pogovora. »Pameten« pa pomeni, da lahko sledi temi, o kateri se trenutno pogovarjamo. Če nas torej zanima kakšna znana oseba in potem rečemo: »Povej mi še kaj več, ve, da mislimo na to osebo. Trenutno se asistent nahaja na skoraj vseh sodobnih napravah. Njegova glavna domača platforma je Google Home. Največja konkurenca sta trenutno Amazon Alexa ter Apple Siri. Izpostavili bomo Alexo, ker je najbolj podobna Google Asistentu. Asistent ima Actions, Alexa ima Skille. Prednost Alexe je, kdaj začne poslušati, in je bolj priročno in naravno reči »Hey Alexa, Kakšno je vreme danes?« kot pa »Ok, Google« in vprašanje. Veliko oseb se odloči za Alexo, ker obstaja tudi veliko naprav, ki lahko komunicirajo z njo. Ko gre za kakovost iskanja in pogovora, pa je Google boljša rešitev [1, 12, 14].

### 4.3 Logika delovanja sistema

Vsak stavek uporabnika gre prek mikrofona ali tipkovnice. Stavek se pošlje Dialogflowu na obdelavo, kjer naredi poizvedbo, če se ujema s kakršnimkoli intentom, ki smo ga ustvarili. Če ga najde, se sproži ustrezni intent, drugače pa se sproži nadomestni, kjer uporabniku sporočimo, da ga ne razumemo. Hkrati je zelo pomembno, kakšen je trenutni kontekst pogovora, in glede na to lahko potem s pomočjo strojnega učenja pove, kateri intent naj se sproži. V intentu se preveri, če je vključen webhook, in v primeru, da ga nima, se naključno izbere eden od odzivov, ki smo ga podali. V primeru, da ima vključen webhook, pa pogleda v mapo »functions«, kjer so v eni datoteki napisani vsi intenti. Za vsak intent je napisano, kaj se mora storiti, npr. klic na API, shranjevanje v bazo ipd. Na koncu mora vedno imeti odziv oz. kaj mora sporočiti uporabniku.



Slika 4.3: Prikaz delovanja sistema

## 4.4 Opis poteka razvoja rešitve

Uporabil sem platformo Dialogflow, ker se uporablja za kreiranje chatbotov za Google asistenta. Dialogflow je super za osnovne potrebe, ker ni treba znati programirati, če hočemo narediti kakšen konkreten produkt, pa omogoča tudi to. Lokalno smo ustvarili projekt v programu WebStorm in si vzeli za zgled projekt Googlovega CodeLabsa Actions 2. Tu se lahko naučiš osnov, kako programirati asistenta.

Vse se lahko programira v eni datoteki, imenovani `index.js`, kasneje se jo da razširiti na več datotek za večjo preglednost projekta. V tej datoteki smo napisali vse intente, ki smo jih kreirali v Dialogflowu. V Dialogflowu je bilo treba kreirati agenta, ki je imel intente, kjer smo vsakemu določili vadbene fraze, na katerih Dialogflow lahko izvaja strojno učenje, da ne bi bilo treba reči točno tistega, kar piše, ampak poskusi najti nekaj podobnega.

Ker imamo datoteko lokalno, jo je treba dodati na strežnik. Lahko bi jo dali na svojega, vendar smo se po Googlovih priporočilih odločili, da jo dodamo na strežnike Firebasea. Na ta način Dialogflow zna pogledati to datoteko. Vključiti pa je bilo treba tudi `fullfilment` pod tistimi intenti, kjer hočemo sprožiti svojo kodo, nato pa še pod nastavitvami Dialogflowa omogočiti `Webhook`. Tu dodamo URL, na katerem je ta `index.js` datoteka. Dialogflow ga potem uporablja. Ko se sproži intent, se pogleda, če je `fullfilment` vključen, če je, potem pogleda na tisti URL, kjer je datoteka, in v njej pogleda, če imamo kakšen intent s tem imenom. Če ga imamo, se izvrši tista koda.

Tu lahko podamo besedilo, kaj naj pove, in če je potrebno, lahko izvršimo API-klic nad našo bazo. V primeru, da nimamo vključenega `fullfilmenta`, pa odgovori naključno z enim od odgovorov, ki smo jih podali temu intentu. Intenti so lahko povezani, povezavo med njimi kreiramo z `input` in `output` konteksti. `Input` kontekst pomeni, da mora prejšnji intent v pogovoru vsebovati ta določen kontekst, če hočemo, da se naslednji intent sproži. Vsak intent lahko vsebuje tudi parametre, ki jih določimo, npr. datume, lokacije, starosti itd. To so entitete, uporabni podatki iz besedila, ki jih ločimo

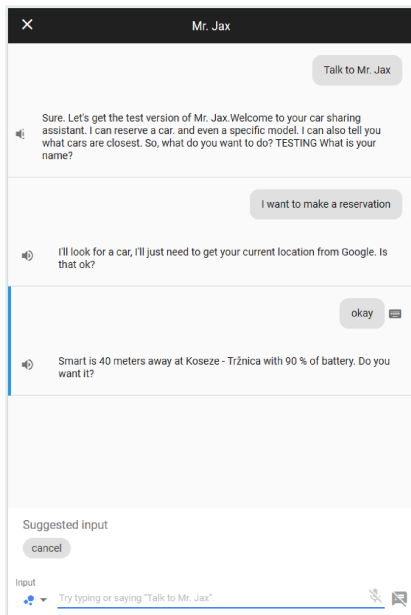
z različnimi tipi in jih lahko tudi kreiramo. V našem primeru smo morali kreirati entiteto modeli avtomobilov in dodati besede, ki imajo ta tip, npr. BMW, Smart, Renault. To so besede, ki bodo poslane naprej. Sprožimo jih lahko s podobnimi besedami, ki jih določimo sami. Primer za Renault: lahko rečemo Zoe in se bo sprožil Renault, ker smo podali Renaultu sinonim Zoe.

## 4.5 Delovanje Dialogflow-a

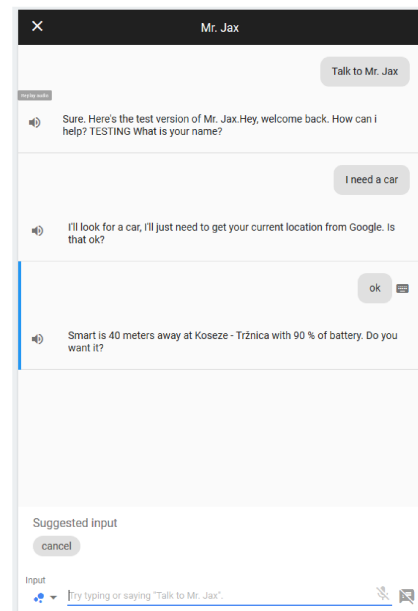
Do sedaj smo opisali celotni sistem, kako je vse povezano. V nadaljevanju bomo razložili, kako deluje Dialogflow. Vsa opažanja in sklepanja so lastna, ker gre za črno škatlo in ne odprtokodni projekt. Dialogflow skrbi za procesiranje naravnega jezika in poveže, kateri intent naj se sproži, mi pa moramo samo napisati, kaj naj se zgodi.

Če uporabnik želi rezervirati vozilo, lahko to naredi na več načinov, kot je prikazano na slikah 4.4 in 4.5. Gre za dva različna stavka, vendar ju Dialogflow vseeno klasificira za enak intent. Stavek se zdi na prvi pogled preprost, vendar bi potrebovali ogromno pogojnih stavkov za implementacijo naravnega jezika. Dialogflow zato uporablja robusten jezikovni interpretator, ki razume, da uporabnik govori enako stvar.

Uporabnika bi lahko tudi s tradicionalnimi računalniškimi vmesniki prisilili, da vnese točno to, kar pričakujemo, s čimer bi se izognili nepričakovanim stavkom. V Dialogflowu se tem predlogom reče »Chips«. Primer »Chipsa« je oblaček »Cancel«, s katerim uporabnik lahko zapre aplikacijo. Dialogflow torej prejme stavek, ga obdela in sproži pravi intent.



Slika 4.4: Primer rezervacije 1



Slika 4.5: Primer rezervacije 2

Prej smo omenili, da je Dialogflow črna škatla. Črna škatla je izraz, ki se uporablja za kodo, v katero pošljemo podatke in nam vrne rezultat, ne da bi vedeli, kako to naredi. Kljub temu smo lahko s pomočjo razhroščevalnika in z beleženjem pretoka podatkov do neke mere raziskali, kako deluje in katere podatke uporablja. V razhroščevalniku se opazi spremenljivka »intentDetectionConfidence«, ki je med 0 in 1, ter predvidevamo, da pomeni, koliko je prepričan, kateri intent bo izbral. Hkrati smo opazili, da je vseeno, če rečemo »cars« ali »car«, razen če imamo točno definirano besedo. Torej lahko sklepamo, da besede, ki imajo enak koren besede, vrže v isti koš. Večjo težo dobijo entitete, ker je s tem potem še bolj prepričan, da gre za ta intent, saj jih imamo v stavkih za usposabljanje določene kot parameter.

Prek dokumentacije lahko prav tako ugotovimo, da sta na izbiro dve možnosti za ujemanje intentov.

**Hibrid** poskuša prek pravil tvorjenja stavkov in training stavkov izbrati pravi intent, drugače pa preklopi na strojno učenje. To je namenjeno za primer, če imamo majhnega agenta, torej do 10 intentov.

**ML klasifikacija** je številka, ki ima razpon med 0,0 (ni prepričan) in 1.0 (popolnoma prepričan). Če pade preveč nizko, gre v nadomestni intent, ki je namenjen večjim agentom.

Dialogflow uporablja tudi preverjanje črkovanja. Če pišemo besedilo in se zatipkamo ali pa namesto 'car' prepozna 'cal', bo vseeno prepoznal, da smo mislili 'car' [8, 18].

## 4.6 Uporabljeni podatki iz API-ja

Pri klicanju na API lahko pridobimo ogromno podatkov o vozilih, uporabniku in lokacijah, vendar ne potrebujemo vseh.

Potrebovali smo samo naslednje JSON-atribute:

- Lokacija:
  - id
  - name
  - distance
  - geoLocation
  
- Vozilo:
  - id
  - name
  - manufacturer
  - seats
  - range
  - batteryCapacity

Lokalno se lahko shranijo podatki o vozilih in lokacijah. Shranjujejo se v obliki JSON in oboje lahko poljubno spreminjamo, dodajamo svoje attribute ipd. Obstajata dva načina lokalnega shranjevanja; s t. i. pogovornim žetonom ali pa neposredno v shrambo uporabnika [23].

**Pogovorni žeton** lahko pokličemo z `conv.data`, kjer je `conv` instanca `Conversation`. S tem lahko shranjujemočasne podatke, saj bodo shranjeni samo, dokler se pogovarjamo z uporabnikom, takoj ko uporabnik zapusti pogovor, pa se podatki pobrišejo. To smo uporabljali za shranjevanje lokacij ter podatkov o tem, za katero lokacijo se uporabnik zanima, saj bodo vedno nove lokacije.

**Uporabniški podatki** `conv.user.storage`, kjer je `conv` instanca `Conversation`. Kar se shrani v shrambo uporabnika, lahko uporabljamo v različnih pogovorih, tudi če uporabnik zapusti pogovor, imamo še vedno shranjene podatke. Izbriše se lahko samo v primeru, da ga zberemo mi, da asistent ne prepozna uporabnika ali pa da uporabnik sam zberse vse podatke o sebi v aplikaciji. S to metodo lahko personaliziramo pogovornega robota, vseeno pa ne smemo shranjevati preveč podatkov, saj je prostor omejen na 10.000 bajtov.

## 4.7 Glasovni pogovorni robot

Ta aplikacija je v osnovi pogovorni robot. To pomeni, da komunicira z uporabnikom v svoji domeni z uporabo naravnega jezika. Obstaja ogromno platform, kjer jih lahko ustvarimo brez kakršnegakoli pisanja kode. To pomeni, da je za točno določeno vprašanje točno določen odgovor (pogojni stavki). To so osnovni pogovorni roboti, ki ne uporabljajo strojnega učenja. Obstajajo tudi pogovorni roboti z umetno inteligenco. Ti imajo integriran NLP oz. naravno procesiranje jezika. To pomeni, da sprocesirajo stavke in izluščijo besede, iz katerih bi lahko razbrali, kaj je uporabnik hotel.

Omogočajo tudi sinonime, ker lahko eno stvar povemo na veliko načinov. Glasovni pogovorni robot je nadgradnja, vendar je primaren način za sporazumevanje glas in ne zaslon. Oboje omogoča, da uporabnik opravi nalogo ali najde informacijo. Velika prednost glasovnega pogovornega robota je dostopnost. To je idealni vmesnik, kjer ne potrebujemo rok, da bi komunicirali z njim, kar pride prav pri kuhanju, ko nam narekuje recept. V tem primeru pride v poštev, če se odpravljamo v službo in potrebujemo vozilo ter lahko med tem, ko se oblačimo, rezerviramo najbližje vozilo [3, 4].

## 4.8 Glasovni uporabniški vmesnik

Aplikacija nima enakega vizualnega vmesnika kot npr. mobilne ali spletne aplikacije, ker se uporablja glas in moramo s pogovorom voditi uporabnika ter s tem oblikovati aplikacijo. Obstaja možnost za prikaz besedila, tabel in gumbov, vendar v večini primerov ne gledamo v zaslon, ampak se samo prostoročno pogovarjamo. To pomeni, da je glas na prvem mestu. Prva uporaba VUI sega v leto 1980, kjer so računalniški sistemi prek telefonov izvrševali ukaze. Trenutno smo v drugem obdobju, kjer se uporablja NLP, ASR – automatic speech recognition ter AI – artificial intelligence. Pri oblikovanju VUI-ja moramo skrbeti, da uporabnika ne zasujemo s ogromno količino aplikacij in da mu ne povemo premalo [24].

## 4.9 Učinkovita komunikacija

Pri učinkovitem komuniciranju obstajajo osnovna štiri načela sodelovanja, ki jih je opisal Paul Grice. Imenujejo se Grice's Maxims.

**Princip kvalitete**, povej resnico.

**Princip kvantitete**, povej dovolj informacij, ne preveč, ker bi preobremenile uporabnika, ter ne premalo, da ve, kako nadaljevati.

**Princip ustreznosti**, povej smiselne informacije.

**Princip obnašanja**, način komunikacije mora biti jasen, torej brez nejasnosti ali dvoumnosti.

Ti principi so zelo pomembni, saj so raziskave pokazale, da ljudje s tehnologijo komuniciramo enako kot s človekom. Ljudje potem pričakujemo, da tudi ko se komunicira z uporabniškim vmesnikom, to poteka enako, kot da bi se pogovarjal z drugim človekom. Z variacijo odzivov pa lahko dosežemo bolj naraven pogovor. Če imamo zmeraj isti odziv, se sliši zelo robotsko in monotono; več kot je variacij, bolje je [5, 13].

## 4.10 Prepoznavanje govora in glasovni pomen

Po zadnjih statistikah je stopnja napake 5% v angleškem jeziku, kar pomeni, da že skoraj vse pravilno prepoznamo in je že skoraj zanemarljivo. Večji problem je glasovni pomen. V večini smo razvili sposobnost razumeti skrita sporočila v pogovoru. Primer: če se pogovarjate in ste žejni in bi hoteli alkohol, bi lahko rekli: »Res potrebujem pijačo.« Prijatelj bi odgovoril: »Si že poskusil pri (IME GOSTILNE).« V preprostem stavku »Res potrebujem pijačo.« je nekaj skritih sporočil: npr. to, da je velika verjetnost, da potrebuje alkoholno pijačo in da je gostilna odprta in blizu. Pogovorni robot bi moral vse to razbrati, zato mu zastavi dodatna vprašanja. To bo prikazano pri naslednjem primeru.

Uporabnik lahko reče »Play me Yesterday.«. S stavkom ni nič narobe, vse je slovnično pravilno, vendar se ne ve, kaj točno hoče, ali hoče igrati igro Yesterday, mogoče je to film ali pa glasba. Prav tako ne ve, od katerega izvajalca je glasba, ker obstaja ogromno pesmi z naslovom »Yesterday«. Zato se postavljajo še dodatna vprašanja, da se ugotovi, kaj uporabnik hoče. V ta namen se trenutno uporabljajo konteksti, da lahko vsaj približno razumemo, kako lahko pogovorni robot pomaga [10].

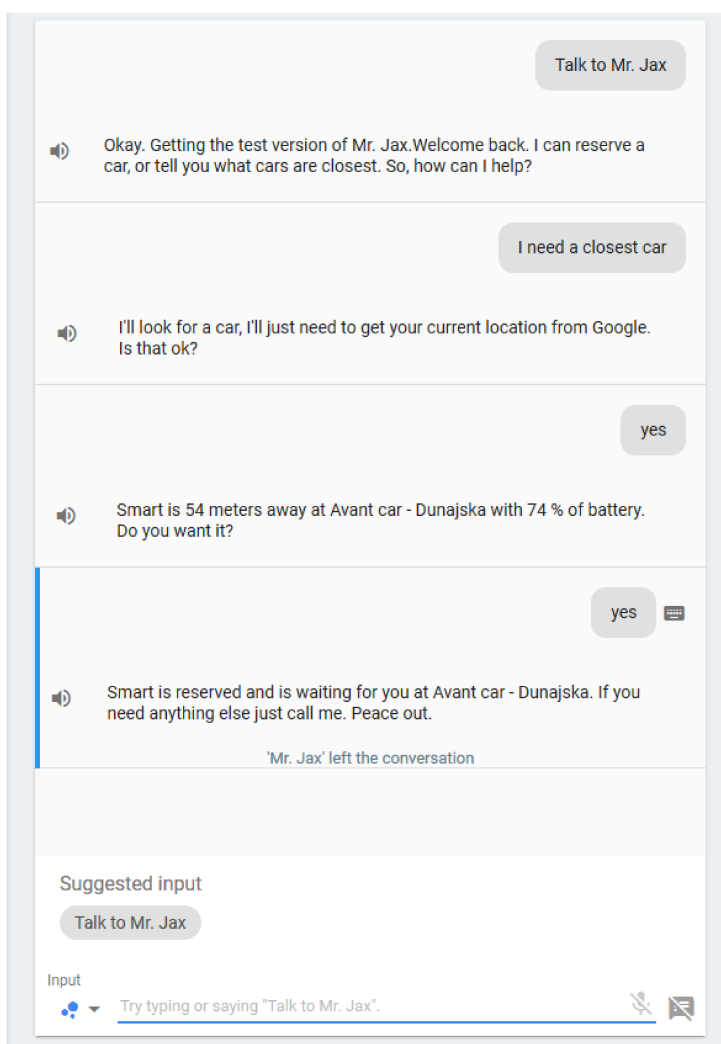
## Poglavje 5

### Ključne funkcionalnosti s primeri srečne poti

Idealni scenarij, kjer uporabnik želeno doseže brez kakršnihkoli problemov, se imenuje »Srečna pot«, kar pomeni, da uporabnik doseže rezultat na najbolj preprost način in brez težav. V nadaljevanju bomo opisali primere funkcionalnosti s pogovornim robotom.

## 5.1 Neposredna rezervacija

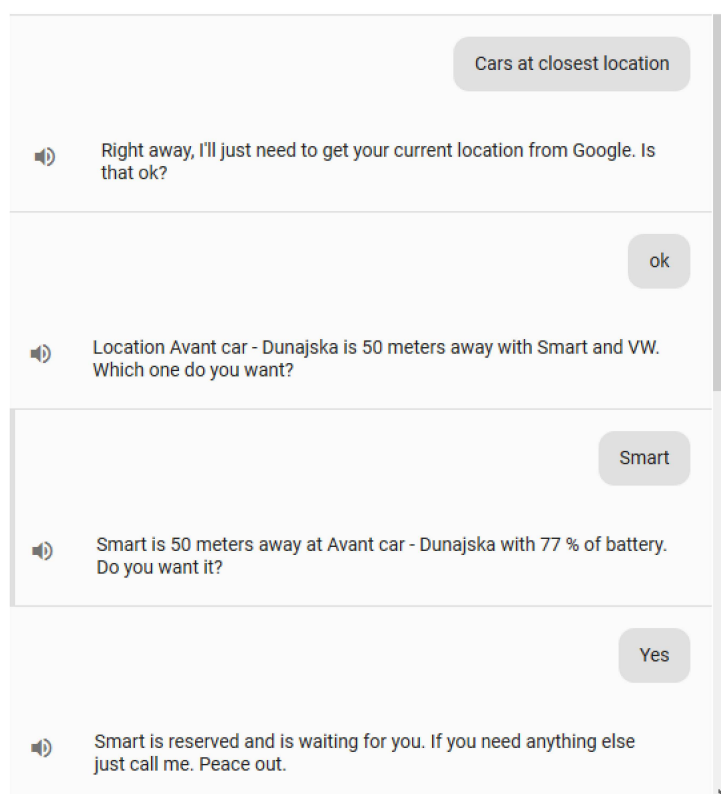
Glavni namen tega projekta je, da lahko uporabnik najhitreje rezervira vozilo. Spodaj je prikazan primer, kako bi ga lahko najhitreje rezerviral. V tem primeru je uporabnik hotel najbližji avto. Ponudil mu je avtomobil Smart in na koncu je samo še potrdil rezervacijo. Na vsakem koraku ponujamo »Chipse«.



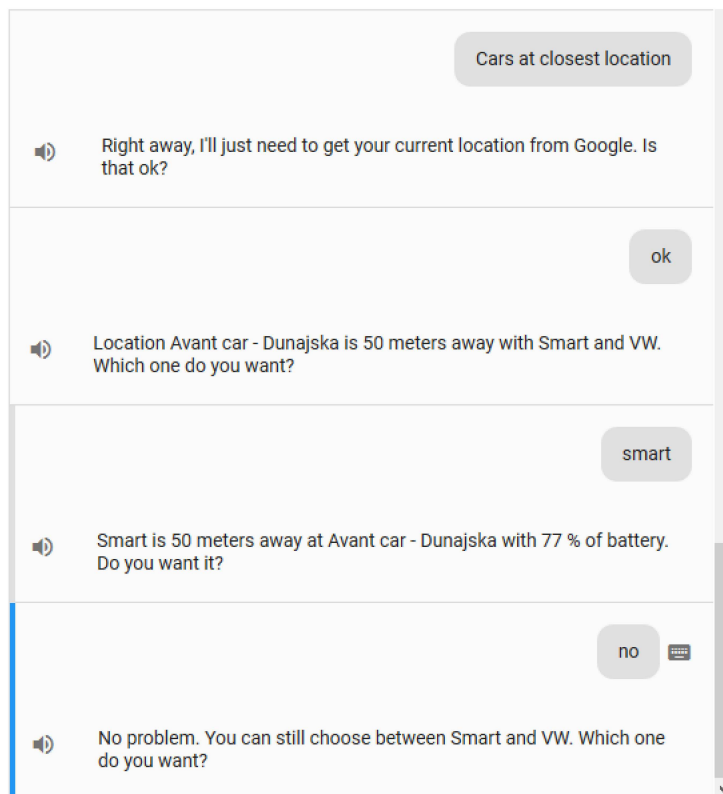
Slika 5.1: Primer neposredne rezervacije

## 5.2 Izbira vozila, glede na lokacijo

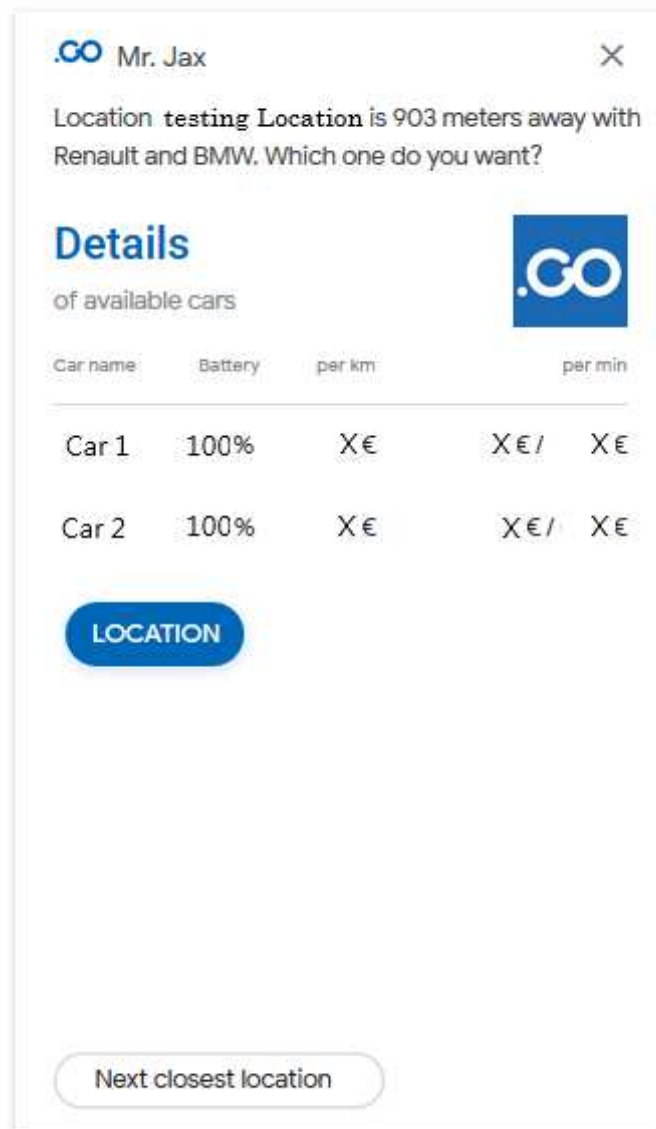
Uporabniku se med pogovorom na zaslonu, če je na voljo, izriše tabela, kot jo prikazuje slika 5.4. V tej tabeli so podrobnosti o vozilu, in sicer cena, koliko še ima baterije in model avtomobila. Vsi trije podatki so ključnega pomena. Cena je zelo pomembna, ker mora biti vse transparentno s plačili, da uporabnik ve, koliko ga bo stalo, ko bo začel z vožnjo. Podatek o bateriji je pomemben, da ve, če je dovolj, da se pripelje do svoje destinacije. Hkrati pa je na voljo gumb »Lokacija«, ki ob pritisku odpre Google Maps in navodila za pot. Pri potrditvi za rezervacijo si lahko uporabnik premisli in pogovorni robot mu bo še enkrat povedal, kaj ima na voljo 5.3.



Slika 5.2: Izbira in potrditev rezervacije



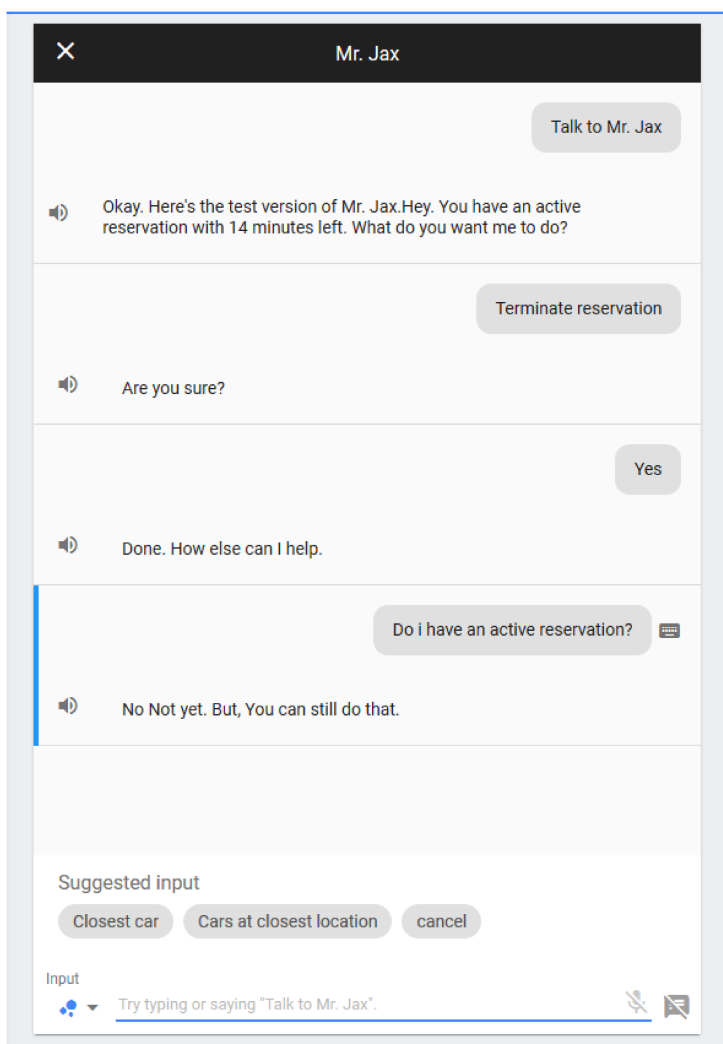
Slika 5.3: Uporabnik si premisli pri potrditvi



Slika 5.4: Vizualni prikaz vozil na voljo

### 5.3 Upravljanje aktivne rezervacije

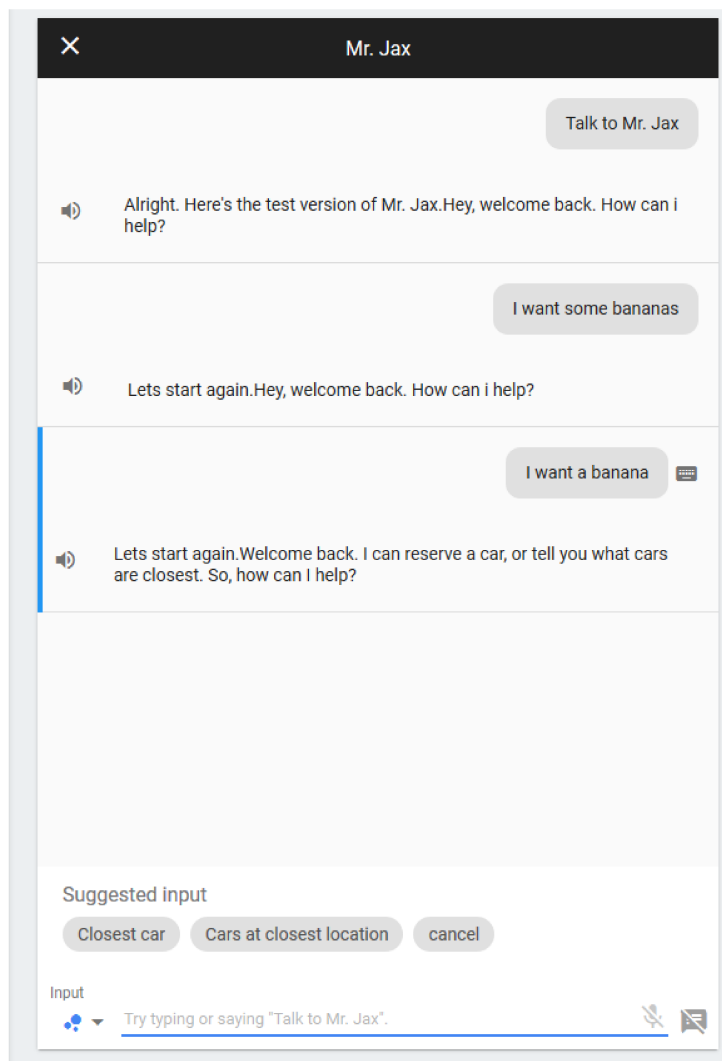
Omogoča tudi preverjanje statusa rezervacije. Uporabnik lahko prekliče rezervacijo ali pa jo podaljša. Za vsako od teh možnosti ga vpraša za potrditev. Ob zagonu aplikacije se preveri, če ima status rezervacije, lahko pa ga vpraša na katerikoli točki pogovora.



Slika 5.5: Primer pogovora z preklicitvijo rezervacije in vprašanjem, če je potem aktivna

## 5.4 Obvladovanje napak med pogovorom

Uporabnik lahko reče nekaj, česar Dialogflow ne bi povezal z nobenim intencijom. V tem primeru uporabniku sporočimo, da ne razumemo, kaj je mislil.



Slika 5.6: Primer pogovora, ko uporabnik reče »banana«



## Poglavje 6

# Možnosti nadaljnjega razvoja in monetizacija

### 6.1 Nadaljnji razvoj

Produkt, ki smo ga razvili, predstavlja osnovo, ki dopušča možnost nadgradenj. To je bil samo začetek aplikacije, možnosti za izboljšave pa iz dneva v dan naraščajo. Google namreč že sam po sebi izboljšuje način govora, dodaja jezike in nam ni potrebno storiti veliko, vseeno pa lahko vnesemo veliko sprememb in dodatnih funkcionalnosti, ki bi dodatno izboljšale izkušnjo.

- Definitivno bi lahko dodali funkcijo »Small Talk«, ki jo omogoča Dialogflow in je privzeto izključena. S tem bi lahko uporabnik asistenta vprašal, kako mu je ime, kako se počuti in razna druga vprašanja, ki niso povezana s funkcionalnostmi aplikacije.
- Integracija z mobilno aplikacijo. Trenutno že obstajata dve mobilni aplikaciji. Dodatna funkcionalnost bi lahko bila, če bi lahko različne zaslone v aplikaciji odprli s pomočjo glasu. Če asistentu rečemo, da bi radi videli svojo rezervacijo v aplikaciji, nas bo vprašal, ali je to ta aplikacija, s katero bi jo rad pogledal. Po potrditvi bi nas preusmerilo na pravi zaslon, kjer bi lahko nadaljevali s pogovorom.

- Statistika in zgodovina opravil. Lahko bi vprašal tudi, koliko smo že zapravili ali koliko denarja še imamo.
- Odklepanje in zaklepanje avtomobila.
- Dodati slovenski jezik.
- Rezervacija vozila z dodatnimi parametri, kjer bi lahko rekli, da hočemo vozilo, ki je za štiri osebe in ima na voljo še več kot 50% baterije.
- Dodatna pomoč pri navigaciji do vozila. Lahko bi rekli, da potrebujemo pomoč do vozila, in bi odprl Google Maps in vklopil navigacijo.
- Opomnik za rezervacijo.
- Med vožnjo naj najde najbližje prosto parkirišče.
- Možnosti shranjevanja najljubših lokacij.
- Še več dodatnih raznoraznih odzivov.
- S pomočjo SSML-ja je mogoče dodati svojo zvočno datoteko, zato bi bilo možno, da se najame zvočnega umetnika in pove vse stavke, jih shranimo na strežnik ter jih potem samo predvajamo.
- Možnost eksplicitne funkcionalnosti pomeni, da bi lahko dostopali do asistentovih intentov, ne da bi bili v aplikaciji. Npr. namesto »Hey Google, talk to Car Sharing« bi lahko rekli »Hey Google, talk to Car Sharing about my reservation«. Na ta način bi aplikacija vedela, da mora povedati njegovo stanje o rezervaciji. Potenciala za razvoj in izboljšave asistenta, s katerimi ga bi naredili pametnejšega, je zelo veliko. Dodali bi lahko odzive za razna vprašanja, ki bi jih imel uporabnik o agentu in akciji, torej Small Talku. Možna je tudi rezervacija specifično določenega avtomobila. Rezervacija vozila glede na parametre, kar pomeni glede na baterijo in ceno. Nato še preverjanje aktivne rezervacije, odklepanje in zaklepanje vozila, opomnik vozila, večjezična podpora za

slovenščino, SSML-implementacija, kar pomeni bolj naravna izgovorjava stavkov in možnost dajanje svojega glasu agentu. Lahko ni našli najbližje parkirno mesto, shranjevali lokacije in ga vprašali, če je na tem mestu kakšno vozilo ali možnost parkiranja.

## 6.2 Možnosti monetizacije

Zasluzek ni neposreden, če izvzamemo podaljsevanje rezervacij, saj to ni asistentova naloga. Asistentova naloga je dostopnost. To, da lahko na vsakem koraku samo z nekaj stavki rezerviramo vozilo, ne da bi odprli aplikacijo, kar je velika konkurenčna prednost, ker je to ena od novejših tehnologij, ki še ni bila uporabljena v Sloveniji. Uporabniki bi prav zaradi te dostopnosti raje uporabljali ta car sharing kot drugega, ker lahko avto rezerviramo z Android, iOS ali pa sedaj z Google Asistentom med tem, ko se nam mudi v službo.



## Poglavje 7

### Zaključek

Namen tega diplomskega dela sta bili poenostavitev in pospešitev procesa najema vozil. Najprej je bilo treba ustvariti načrt uporabe, kjer smo definiralo primere uporabe in vizualizirali tok pogovora. Opisali smo tehnologije, ki so pripomogle k izdelavi pogovornega robota. Ustvarjen je bil z osnovnimi funkcionalnostmi, ki nam omogočajo rezervacijo vozila s pomočjo glasu v Google Asistentu. Ustvarjeni so bili intenti za neposredno rezervacijo, ogled najbližje lokacije in informacije o trenutni rezervaciji. Na koncu smo razložili, kako ga bi lahko izboljšali še z dodatnimi funkcionalnostmi, kot je npr. odklepanje vozila, in opisali, na kakšen način bi lahko ustvarili prihodek.



# Literatura

- [1] Alexa. Dosegljivo: <https://voicebot.ai/2018/09/02/amazon-alexa-now-has-50000-skills-worldwide-is-on-20000-devices-used-by-3500-brands/>, 2019. [Dostopano 16. 1. 2019].
- [2] Axios. Dosegljivo: <https://github.com/axios/axios>, 2018. [Dostopano 5. 10. 2018].
- [3] Types of bots. Dosegljivo: <https://medium.com/botsupply/rule-based-bots-vs-ai-bots-b60cdb786ffa>, 2018. [Dostopano 13. 12. 2018].
- [4] Chatbots reasons to use the platform. Dosegljivo: <https://www.rootinfosol.com/top-6-reasons-to-use-google-chatbot-platform-dialogflow>, 2018. [Dostopano 10. 12. 2018].
- [5] Conversation design. Dosegljivo: <https://designguidelines.withgoogle.com/conversation/conversation-design/learn-about-conversation.html>, 2018. [Dostopano 14. 11. 2018].
- [6] CIP Automotive Car Sharing in Europe. Dosegljivo: <https://www2.deloitte.com/content/dam/Deloitte/de/Documents/consumer-industrial-products/CIP-Automotive-Car-Sharing-in-Europe.pdf>, 2018. [Dostopano 19. 9. 2018].
- [7] Dialogflow features. Dosegljivo: <https://blog.dialogflow.com/post/apiai-new-name-dialogflow-new-features/>, 2018. [Dostopano 10. 11. 2018].

- 
- [8] Dialogflow intro. Dosegljivo: <https://dialogflow.com/docs/intro/>, 2018. [Dostopano 23. 9. 2018].
- [9] Firebase functions. Dosegljivo: <https://firebase.google.com/docs/functions/>, 2018. [Dostopano 9. 9. 2018].
- [10] Future of speech recognition. Dosegljivo: <https://www.globalme.net/blog/the-present-future-of-speech-recognition>, 2018. [Dostopano 7. 9. 2018].
- [11] Google I/O '17. Dosegljivo: [https://www.youtube.com/watch?v=wuDP\\_eygsvs&list=PLJ21zHI2TNh9VkAu1Es0hpw92Wkm-XcuD](https://www.youtube.com/watch?v=wuDP_eygsvs&list=PLJ21zHI2TNh9VkAu1Es0hpw92Wkm-XcuD), 2018. [Dostopano 7. 10. 2018].
- [12] Comparison with Google Assistant. Dosegljivo: <https://www.trustedreviews.com/opinion/google-assistant-vs-alexa-3370865>, 2019. [Dostopano 16. 1. 2019].
- [13] H Paul Grice. *Logic and conversation*. 1975, pages 41–58, 1975.
- [14] How assistant works. Dosegljivo: <https://codeburst.io/2-how-assistant-work-introduction-to-dialogflow-319a72ba2db>, 2018. [Dostopano 16. 10. 2018].
- [15] Voice assistant used in Smart homes. Dosegljivo: <https://www.juniperresearch.com/press/press-releases/voice-assistants-used-in-smart-homes>, 2018. [Dostopano 7. 11. 2018].
- [16] Dialogflow languages. Dosegljivo: <https://dialogflow.com/docs/reference/language>, 2018. [Dostopano 18. 10. 2018].
- [17] Google Assistant languages. Dosegljivo: <https://www.androidcentral.com/google-assistant-will-expand-38-countries-and-17-languages-2018-0>, 2018. [Dostopano 29. 11. 2018].

- 
- [18] Machine Learning in Dialogflow. Dosegljivo: <https://miningbusinessdata.com/dialogflow-machine-learning-algorithm/>, 2018. [Dostopano 7. 10. 2018].
- [19] Ngrok. Dosegljivo: <https://ngrok.com/docs>, 2018. [Dostopano 11. 11. 2018].
- [20] NodeJS v2. Dosegljivo: <https://developers.google.com/actions/reference/nodejsv2/overview>, 2018. [Dostopano 17. 11. 2018].
- [21] Promise mozilla. Dosegljivo: [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Promise](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise), 2018. [Dostopano 19. 12. 2018].
- [22] Promise JS. Dosegljivo: <https://www.promisejs.org/>, 2018. [Dostopano 4. 10. 2018].
- [23] Data managment. Dosegljivo: <https://developers.google.com/actions/assistant/save-data>, 2019. [Dostopano 8. 10. 2018].
- [24] VUI. Dosegljivo: <https://searcherp.techtarget.com/definition/voice-user-interface-VUI>, 2018. [Dostopano 28. 9. 2018].
- [25] SSML. Dosegljivo: <https://developers.google.com/actions/reference/ssml>, 2018. [Dostopano 15. 12. 2018].
- [26] Voice search statistics. Dosegljivo: <https://quoracreative.com/article/voice-search-statistics-trends>, 2018. [Dostopano 4. 9. 2018].
- [27] Speech synthesis. Dosegljivo: <https://www.w3.org/TR/speech-synthesis/>, 2018. [Dostopano 17. 12. 2018].
- [28] Webstorm. Dosegljivo: <https://www.jetbrains.com/webstorm/>, 2019. [Dostopano 15. 1. 2019].