# Razgovorni agent u pametnoj okolini Interneta stvari

**Marinello, Domagoj**

SVEUČILIŠTE U ZAGREBU
**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

DIPLOMSKI RAD br. 430

# RAZGOVORNI AGENT U PAMETNOJ OKOLINI INTERNETA STVARI

Domagoj Marinello

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

DIPLOMSKI RAD br. 430

# RAZGOVORNI AGENT U PAMETNOJ OKOLINI INTERNETA STVARI

Domagoj Marinello

Zagreb, lipanj 2024.

**SVEUČILIŠTE U ZAGREBU**
**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

Zagreb, 4. ožujka 2024.

# DIPLOMSKI ZADATAK br. 430

| | |
|---|---|
| Pristupnik: | **Domagoj Marinello (0036507649)** |
| Studij: | Računarstvo |
| Profil: | Znanost o podacima |
| Mentor: | izv. prof. dr. sc. Marin Vuković |

Zadatak: **Razgovorni agent u pametnoj okolini Interneta stvari**

Opis zadatka:

Pametne okoline Interneta stvari (IoT) mogu se definirati kao fizički prostori obogaćeni senzorima, aktuatorima, sučeljima za interakciju i raznim uslugama koje mogu biti samostalne ili ugrađene u uređaje. Elementi pametne okoline tipično su povezani na IoT platformu koja omogućava nadzor i upravljanje. Razgovorni agent je računalni program koji ostvaruje interakciju s čovjekom korištenjem prirodnog jezika, a može se primijeniti u širokom području ljudskih djelatnosti. Vaš zadatak je implementirati razgovornog agenta koji se izvodi u pametnoj okolini Interneta stvari i pruža informacije o stanju u okolini na zahtjev korisnika, a omogućava i postavljanje stanja na uređajima dostupnim u pametnoj okolini. Interakcija s razgovornim agentom ostvaruje se korištenjem hrvatskog jezika. U okviru zadatka potrebno je definirati i kako bi se razgovorni agent integrirao sa uslugama koje pružaju funkcionalnosti prepoznavanja govora i strojne tvorbe govora.

Rok za predaju rada: 28. lipnja 2024.

Htio bih se zahvaliti svojim najbližim prijateljima i obitelji koji su mi uvijek bili oslonac kroz studiranje. Posebne zahvale idu mojoj boljoj polovici Lucii.

## Sadržaj

# Uvod

Razgovorni agenti i upiti koje oni mogu zadovoljiti jedna su od najpopularnijih tema u današnjem svijetu. Razni modeli za razgovor doveli su do naprednih mogućnosti obrade prirodnog jezika i procesiranja velikih količina teksta.

Jedna od popularnih primjena razgovornih agenata je u obliku asistenta. Agent asistent je prilagođena verzija agenta koja se primjenjuje za podršku korisnicima u nekom specifičnom području rada. Razgovornim sposobnostima agenta se pospješuje prikupljanje informacija u zadanom području, a agentu se može omogućiti korištenje digitalnih alata kako bi bolje radio svoj „posao".

Ovaj diplomski rad opisuje izradu razgovornog agenta u IoT (eng. Internet of Things) okruženju. Ovakav agent ima mogućnost izvještavanja korisnika o stanju njegovih povezanih uređaja, predlaže radnje koje je potrebno izvršiti te može razgovarati s korisnikom. U okviru rada ispitane su agentove sposobnosti te je konačno donesen zaključak o agentovoj uspješnosti.

# 1. Pregled područja

U ovom poglavlju opisan je pregled područja razgovornih agenata. Definiran je pojam razgovornog agenta, opisane su moguće primjene te primjeri takvih agenata.

## 1.1. Pojam razgovornog agenta

Razgovorni agent može se definirati kao računalni program koji koristi umjetnu inteligenciju kako bi odgovorio na zahtjeve korisnika. On pri tom može koristiti određene alate koji su mu omogućeni kako bi bolje odgovorio na korisnikov zahtjev ili upit [1].

Jedna od preciznijih definicija razgovornih agenata ih definira kao računalne programe dizajnirane za prirodan razgovor s ljudskim korisnicima, bilo da se radi o neformalnom razgovoru, u kojem slučaju se sustav obično naziva *chatbotom*, ili o razgovoru s ciljem pružanja relevantnih informacija vezanih uz određeni zadatak, u kojem slučaju se sustav naziva agentom usmjerenim na zadatak [2]. Agent usmjeren na zadatak pružanja izvještaja IoT okoline je tema ovog diplomskog rada.

### 1.1.1. Vrste razgovornih agenata

Danas pojam razgovornog agenta često podrazumijeva agenta koji radi uz pomoć modela strojnog učenja s kojim procesira ulazne informacije. Osim takvih, postoje i agenti koji se temelje na sustavu pravila. Oni procesiraju korisnički ulaz primjenjujući pravila, a ako za neki upit nema primjenjivog pravila korisnik će dobiti odgovor koji mu vjerojatno nije koristan [1].

Neki konverzacijski agenti komuniciraju isključivo putem teksta, dok drugi uključuju složenije oblike unosa i izlaza poput govora. Postoje i takozvani utjelovljeni razgovorni agenti koji su obično opremljeni animiranom vizualnom reprezentacijom na ekranu [2].

Vjerojatno najvažniji dio konverzacijskih agenata je način na koji predstavljaju i obrađuju dijalog. I tu postoji širok raspon mogućnosti, od jednostavnih sustava temeljenih na podudaranju predložaka do vrlo složenih reprezentacija temeljenih na dubokim neuronskim mrežama [2].

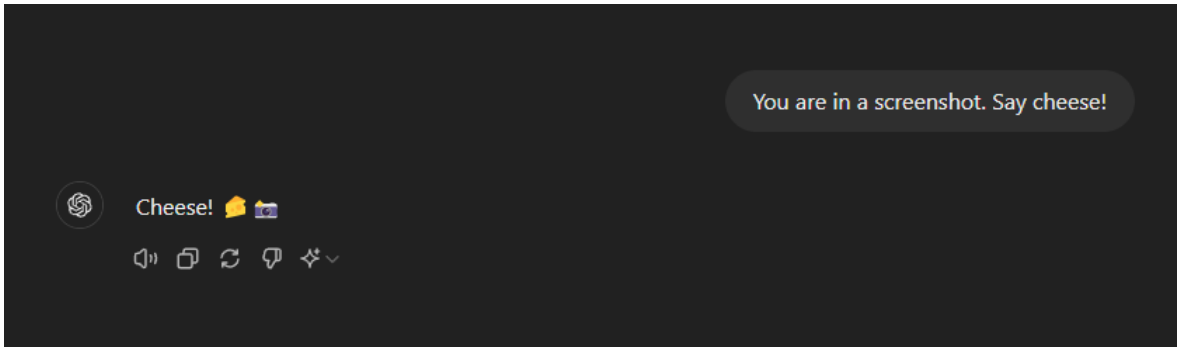
## 1.2. Primjene razgovornih agenata

Razgovorni agenti se mogu primjenjivati u područjima koja zahtijevaju jednostavnu ljudsku interakciju. Moguće je, na primjer, stvoriti virtualnog agenta koji će služiti za pružanje korisničke podrške.

### 1.2.1. Općenito

Moguća je izrada agenta za opći razgovor. Najpoznatiji primjer takvog agenta je ChatGPT od OpenAI-a. Radi se o *chatbotu* koji na korisničke upite odgovara u prirodnom jeziku, a sposoban je i za pretraživanje interneta te generiranje programskog koda [3].

Primjer interakcije s ChatGPT-em prikazan je na slici (Sl. 1.1).




Sl. 1.1 Interakcija s ChatGPT-em

Prikazana je šaljiva interakcija s agentom kojem je rečeno da će biti uslikan. Agent je shvatio ton interakcije te je odgovorio na prikladan način. Zadovoljene su eksplicitne upute, „Say cheese!”, kao i implicitne upute. Implicitne upute u ovom slučaju bi bile „Ponašaj se kao da te zapravo fotografiram”.

Agent može vrlo lako zadovoljiti ovako jednostavan upit. Kada je od agenta potreban jako specifičan odgovor ili obrada podataka, onda je potrebno zadati detaljan upit. Izrada i unaprjeđivanje upita pokrivena je novonastalim područjem koje se zove inženjerstvo upita (eng. prompt engineering).

### 1.2.2. Zdravstvo

Razgovorni agenti mogu igrati ključnu ulogu u zdravstvu i medicinskoj skrbi. Oni mogu pomoći liječnicima tijekom konzultacija, podržavati korisnike u promjenama ponašanja ili pomagati pacijentima i starijim osobama u njihovim domovima. Iako ovi razgovorni agenti nude velike prednosti, prisutni su i potencijalni sigurnosni rizici koji mogu ugroziti

pacijente kao što su nepoštivanje privatnosti, tehnički problemi, štetni odgovori i tehnički problemi [4]. Navedeni problemi možda objašnjavaju zašto je upotreba razgovornih agenata u zdravstvu relativno rijetka.

Jedna od mogućih primjena u zdravstvu su agenti za pomoć pacijentima koji pate od mentalnih oboljenja. Razgovorni agenti imaju značajan potencijal u području liječenja mentalnih poremećaja, posebno u psihološkom obrazovanju i poticanju pridržavanja terapiji. Ti agenti mogu pružati prilagođeni obrazovni sadržaj, pomažući pojedincima da bolje razumiju i lakše prihvaćaju svoje stanje, a osim toga nude i podrške u edukaciji te potiču pridržavanje terapiji pružanjem kontinuiranih podsjetnika na provođenje terapijskih aktivnosti ili uzimanje lijekova [5].

Jedan primjer takvog agenta je agent slušatelj (eng. listener) koji pomaže u socijalizaciji ljudi koji pate demencije, a služi im kao podrška u razgovoru [6]. Agent je dizajniran za interakciju s pacijentima putem postavljanja pitanja i pružanja povratnih informacija tijekom razgovora, te snimanja govornih i jezičnih podataka radi procjene kognitivnog statusa [6]. Ovakvi agenti imaju pozitivan utjecaj posebno onda kada nema dovoljno ljudskih resursa za pomoć pri psihološkim terapijama. Dodatno, pacijenti su pokazali veći angažman kada je agent pružao povratne informacije tijekom razgovora što ističe preferenciju za razgovorne agente koji pokazuju društvene signale [6].

Drugi primjeri su robot PARO i agent Woebot. PARO je fizički agent u obliku tuljana, a svrha mu je da bude ljubimac ljudima kojima nedostaje socijalizacije i fizičkog kontakta, a posebno je pogodan za ljude s alergijama koji ne bi mogli imati stvarnog ljubimca s krznom [7]. Iako nije striktno razgovorni agent, PARO omogućuje drugu vrstu komunikacije te je stoga prikladan za razmatranje u ovom kontekstu. S druge strane, Woebot je razgovorni agent koji služi pacijentima mentalnih oboljenja i onima koji nisu dijagnosticirani kao savjetnik i pomaže im da vode brigu o svom mentalnom zdravlju [8].

### 1.2.3. Obrazovanje

Razgovorni agenti imaju sve veću ulogu u obrazovanju. Oni mogu prilagoditi iskustvo učenja za pojedine studente, poboljšati rezultate učenja i podržati nastavnike, posebno na velikim predavanjima i masovnim online tečajevima gdje pojedinačna podrška za svakog studenta nije moguća zbog financijskih i organizacijskih ograničenja [9]. Kvaliteta ovakvih pristupa ovisi o kvaliteti samih agenata, ali i o sposobnostima pojedinih studenata.

Jedan primjer edukativnog razgovornog agenta je nastavnica Sara (eng. Sara, the Lecturer), koja je dizajnirana za unapređenje online video predavanja poticanjem značajnih interakcija sa studentima [10]. Sara koristi tri ključna dizajnerska načela: glavni i pod-dijalog, odgovarajuće metode dijagnostike i adresiranje više kanala. Implementacija ovih načela značajno poboljšava zadržavanje informacija kod učenika i njihovu sposobnost primjene znanja [10].

Sarino prvo načelo vodi razgovor kroz jedan glavni dijalog koji se po potrebi prekida dodavanjem pod-dijaloga koji se sastoji od pitanja studentima, a ovisno o odgovorima, Sara može ili nastaviti glavni dijalog ili pobliže objasniti gradivo s kojima studenti imaju poteškoća [10].

Odgovori se klasificiraju drugim dizajnerskim načelom. Konkretna metoda klasifikacije je NLP (eng. Natural Language Processing) alat koji analizira mišljenje odgovora (eng. sentiment analysis), a rezultat analize omogućuje Sari da odredi hoće li nastaviti glavni dijalog ili će pobliže objasniti gradivo [10].

Treće dizajnersko načelo tiče se načina Sarine komunikacije. Sara svoja pitanja i objašnjenja zadaje tekstom i govorom, a takva se kombinacija komunikacijskih kanala pokazala boljom za zadržavanje znanja u usporedbi s korištenjem čistog teksta [10]. Ovaj diplomski rad će također omogućiti više kanala komunikacije implementacijom podrške za ljudski govor.

Neki drugi primjeri obrazovnih agenata su AutoTutor, Teacherbot i Freudbot. Navedeni agenti su dizajnirani kao potpora obrazovanju. AutoTutor se bavi općim podučavanjem studenata i ispitivanjem njihovog znanja [11]. Teacherbot je koristio tadašnji Twitter kako bi objavljivao obrazovni sadržaj i odgovore na pitanja [12], a agent Freudbot je nastojao približiti gradivo tako što je glumio povijesnu ličnost Freuda dok je objašnjavao gradivo [13].

### 1.2.4. Poslovne primjene

Razgovorni agenti se često koriste u području korisničke podrške za razne tvrtke. Pri tom se pokazalo da nije potrebno skrivati činjenicu da se radi virtualnom agentu, a ne o stvarnoj osobi [14]. Ovo ukazuje na važnost toga da se agenti ponašaju ljudski bez obzira na to jesu li predstavljeni kao ljudi ili kao virtualni agenti.

Jedno istraživanje koje se bavilo sustavom koji pruža korisničku podršku preko društvenih mreža je navelo veliku prednost generativnih modela naspram onih koji samo dohvaćaju podatke [15]. Pokazalo se da generativni neuronski modeli nadmašuju modele temeljene na preuzimanju podataka, no suočavaju se s poteškoćama kada je u skupu za treniranje prisutan vrlo mali broj primjera za određenu temu [15]. Jedan način umanjivanje ove slabosti, je proširenje i diversifikacija skup podataka za treniranje, što omogućuje modelima da bolje uče i generaliziraju u kontekstima s manje informacija. Rezultati ovog istraživanja mogli bi ukazivati na inherentnu prednost agenata zasnovanim na generativnim modelima koji komuniciraju prirodnim jezikom u usporedbi sa starijim metodama.

Istraživanje o ponašanju kupaca luksuznih brendova je pokazalo da zadovoljstvo kupaca s razgovornim agentima luksuznih brendova ovisi o percepciji kvalitetne komunikacije [16]. Prema tom istraživanju, asistenti pri kupnji mogu izgraditi pozitivne odnose s kupcima ako komuniciraju dovoljno dobro da ih se može usporediti s ljudskim asistentima.

Jedan primjer agenta koji se može koristiti za poslovne primjene je „Calendar.help“, ili Cal, sustav koji pruža mogućnost slaganja rasporeda putem strukturiranih radnih tokova [17]. Korisnici komuniciraju s Calom kao s osobnim asistentom, pri čemu najizazovniji zadaci vezani uz slaganje kalendara odlaze obučenim asistentima dok se za uobičajene „*mikro*“ zadatke koriste automatizirani procesi kad god je to moguće [17]. Ovakav sustav omogućava korisnicima da lako delegiraju razne kompleksne zadatke agentima koji koriste strojno i ljudsko znanje različitih razina stručnosti.

### 1.2.5. Kultura

Razgovorni agenti se mogu koristiti za tumačenje kulturne baštine raznim vrstama korisnika. Ovakvi agenti mogu biti korišteni i u svrhe turizma, a u budućnosti bi se često mogli naći u muzejima, galerijama i knjižnicama.

Jedan takav razgovorni agent već postoji. Razgovorni agent muzeja, Mario Praz, koristi popularno sučelje za chat unutar aplikacija za razmjenu poruka bez potrebe otvaranje internet preglednika ili instaliranja muzejske aplikacije [18]. Jednostavnost ovakvog pristupa bi mogla učiniti ovu tehnologiju privlačnijom široj publici.

Drugi primjeri su CulturalERICA agent za istraživanje o europskoj povijesti te Ada i Grace agenti za interakciju s posjetiteljima muzeja. CulturalERICA pruža mogućnost komunikacije uz odgovaranje putem govora i video sadržaja, dok same odgovore pribavlja

putem baze „Europeana“ [19]. Ada i Grace su razgovorni agenti u obliku informativnog kioska koji prepoznaje ljudski govor i pruža odgovore na pitanja na način da ih dohvaća iz baze već definiranih odgovora [20]. Dok je prepoznavanje ljudskog govora kompleksan zadatak, dohvaćanje već definiranih odgovora je danas već zastarjela praksa.

### 1.2.6. Primjena u okolini Interneta stvari

Ovaj diplomski rad bavi se primjenom razgovornih agenata u okolini Interneta stvari. Internet stvari (eng. IoT, Internet of Things) označava mrežu fizičkih uređaja, kućanskih aparata i drugih predmeta opremljenih elektronikom, softverom, senzorima i sposobnošću povezivanja koja im omogućava prikupljanje i razmjenu podataka.

IoT okolina podrazumijeva okolinu u kojoj se nalaze umreženi uređaji. Oni se često povezuju na zajedničku softversku infrastrukturu koja omogućava povezivanje, upravljanje i nadzor IoT uređaja i njihovih podataka. Takva vrsta softvera zove se IoT platforma i ona može razgovornim agentima pružiti informacije potrebne za zadovoljavanje korisničkih zahtjeva.

Tako bi jedna od mogućih primjena u ovom području bio razgovorni agent za generiranje izvještaja o stanju IoT okoline. Takav agent bi od platforme tražio informacije o trenutnom stanju povezanih uređaja, a potom bi korisniku na temelju tih informacija generirao formatirani izvještaj. Implementacija takvog agenta opisana je u sljedećem poglavlju.

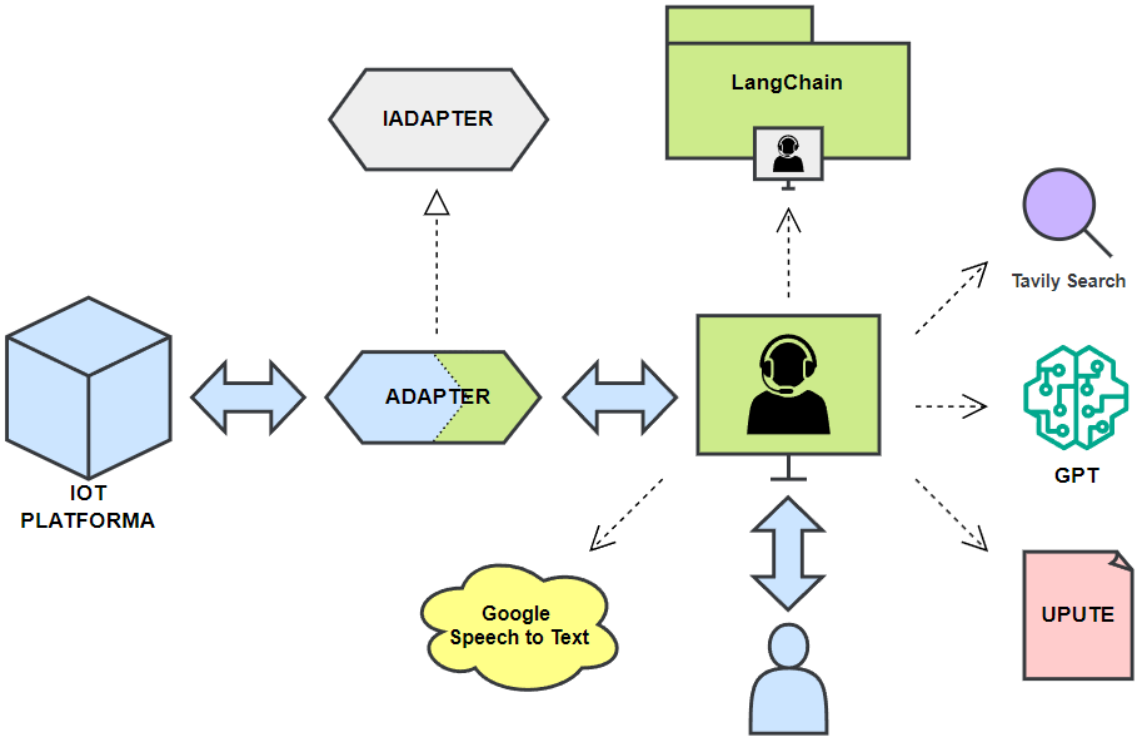
# 2. Implementacija razgovornog agenta

U razvoju agenta koriste se tehnike poput inženjerstva upita. Međutim, razvoj agenta predstavlja jednu višu razinu apstrakcije od toga. Razgovorni agent može sadržavati više parametara modifikacije upita koji se primjenjuju u različitim situacijama, a agent sam odlučuje o koracima koje je potrebno primijeniti kako bi se zadovoljio upit [21].

Ovo poglavlje navodi ključne komponente korištene u izradi i testiranju razgovornog agenta. Agent je izrađen kao samostalna aplikacija u programskom jeziku Python, a neka od ispitivanja su provedena u Jupyter bilježnici u okruženju Google Colab [22].

## 2.1. Arhitektura sustava

Ideja ovog sustava je korištenje razgovornog agenta za dobivanje informacija o stanju IoT okoline. Kako bi mogao pružiti te informacije, agent mora komunicirati s IoT platformom i koristiti više alata koje ima na raspolaganju. Idejna arhitektura sustava prikazana je na slici (Sl. 2.1).



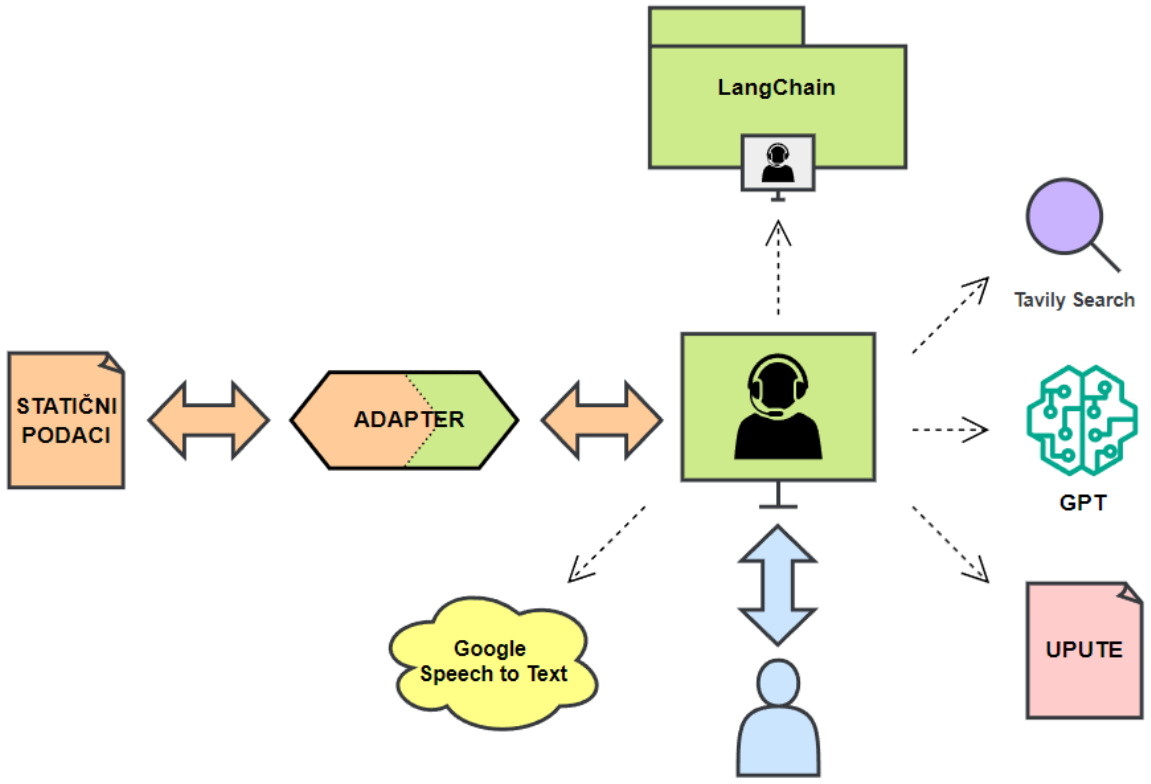

Sl. 2.1 Idejna arhitektura sustava

Arhitektura na slici prikazuje agenta koji s IoT platformom komunicira putem adaptera. Adapter je izveden od sučelja IAdapter kako bi agentu omogućio komunikaciju s tom specifičnom IoT platformom.

LangChain biblioteka omogućava instanciranje agenta uz zadane alate koji su prikazani na desnoj strani. To su alati za pretragu interneta, model za procesiranje prirodnog jezika te upute koje sadrže instrukcije i format upita.

Koristeći dostupne alate, agent odgovara na korisnikove upite, a uz to može koristiti i Google Speech to Text kako bi procesirao upit zadan ljudskim govorom [23].

Ovakvo rješenje podrazumijeva izvođenje u okolini koja sadrži IoT platformu. To nije bilo praktično za testiranje agentovih mogućnosti te je stoga implementirano rješenje sa simuliranom okolinom (Sl. 2.2).




Sl. 2.2 Stvarna arhitektura sustava

Slika prikazuje stvarnu arhitekturu implementiranog sustava. Komunikacija s IoT platformom je simulirana koristeći statične podatke. Adapter ne koristi posebno sučelje jer je predviđen samo za rad u simuliranom okruženju. Korištenje statičnih podataka omogućuje jednostavno i konzistentno testiranje agenta.

## 2.2. Prompt engineering

*Prompt engineering* je proces oblikovanja i optimizacije upita koji se koriste za interakciju s modelima umjetne inteligencije, posebice generativnim jezičnim modelima poput GPT-a (eng. Generative Pre-trained Transformer). Cilj ovog procesa je poboljšati kvalitetu i relevantnost odgovora koje model daje na postavljene upite. *Prompt engineering* uključuje razumijevanje mogućnosti i ograničenja modela, kao i pažljivo oblikovanje upita kako bi se smanjile nejasnoće i povećala učinkovitost komunikacije [24].

### 2.2.1. Tehnike oblikovanja

Oblikovanjem upita daje se jasnoća, specifičnost i kontekst upitima. Neke od tehnika su korištenje primjera, prikaz toka misli, prikaz stabla misli te uključivanje domenskog znanja [25].

Strategije inženjerstva upita mogu uključivati odabir specifičnih riječi ili fraza koje su sklonije generiranju korisnih odgovora, strukturiranje upita na način koji modelima olakšava razumijevanje konteksta i željenih informacija, te prilagođavanje upita specifičnostima pojedinog modela. Ovaj pristup omogućava istraživačima i korisnicima da efikasnije iskoriste napredne AI tehnologije, što dovodi do boljih rezultata i učinkovitije upotrebe u različitim aplikacijama [26].

### 2.2.2. Izrada razgovornog agenta

U kontekstu stvaranja razgovornog agenta, *prompt engineering* podrazumijeva oblikovanje upita ili pitanja koja efikasno usmjeravaju jezični model prema generiranju odgovora usklađenih sa željenim ponašanjem agenta. To uključuje stvaranje jasnih i specifičnih upita, pružanje relevantnog konteksta, uključivanje primjera i iterativno poboljšavanje upita kako bi se osiguralo da razgovorni agent razumije i generira odgovore koji su prikladni. Cilj je usmjeriti jezični model prema stvaranju izlaza koji poboljšavaju sposobnost razgovornog agenta za inteligentnu i učinkovitu interakciju s korisnicima [27].

## 2.3. Radni okvir LangChain

LangChain je platforma za razvoj aplikacija koje koriste jezične modele. Pruža kontekst aplikacijama, povezujući jezični model s izvorima konteksta poput uputa, primjera interakcija (eng. few-shot examples), sadržaja za prilagodbu odgovora itd. [28]. Također podržava razmišljanje, gdje se oslanja na jezični model za razlučivanje kako odgovoriti na temelju pruženog konteksta i donijeti odluku o akcijama [28].

Sastoji se od nekoliko dijelova:

- **LangChain biblioteke:** Python i JavaScript knjižnice koje sadrže sučelja i integracije za razne komponente, osnovni runtime za kombiniranje tih komponenata u lance i agente, te gotove implementacije lanaca i agenata.
- **LangChain predlošci:** Kolekcija referentnih arhitektura za različite zadatke.
- **LangServe:** Knjižnica za implementaciju LangChain lanaca kao REST API-ja.
- **LangSmith:** Razvojna platforma koja vam omogućuje ispravljanje pogrešaka, testiranje, evaluaciju i praćenje lanaca izgrađenih na bilo kojem jezičnom modelu i jednostavno se integrira s LangChain-om.

Od navedenih stavki, za potrebe IoT agenta, bila je potrebna samo implementacija LangChain biblioteke. Integracija u simulator (Poglavlje 0) izvedena je u par linija koda, što pokazuje koliko je biblioteka jednostavna za korištenje (Kôd 2.1)

```
...
llm = ChatOpenAI(model="gpt-3.5-turbo", temperature=0)

search = TavilySearchResults()
tools = [search]

prompt = ChatPromptTemplate.from_messages([...])

agent = create_openai_functions_agent(llm, tools, prompt)
self.agent_executor = AgentExecutor(agent=agent, tools=tools,
verbose=False)
...
```

Kôd 2.1 Jednostavna integracija biblioteke LangChain

Isječak koda prikazuje kôd kojim je integrirana biblioteka LangChain. Model koji se koristi za procesiranje upita je GPT 3.5 Turbo [29]. Na slici je vidljiva integracija alata

Tavily koji će biti detaljnije objašnjen u jednom od sljedećih potpoglavlja. Osim toga, vidljiva je i varijabla upita - *prompt*. Sadržaj varijable nije prikazan u potpunosti, a u njemu se odvija *prompt engineering* uz pomoć uputa modelu koje će biti objašnjene u sljedećem poglavlju.

## 2.4. Podrška za pribavljanje ažurnih informacija

Kako bi korisnik mogao zadavati upite koji traže informacije o trenutnim događanjima u svijetu, agent mora imati sposobnost pretraživanja interneta. Tu sposobnost osigurava alat Tavily Search Results [30].

Integracijom u agenta, ovaj alat pruža modelu rezultate pretrage interneta po korisničkom upitu. Agent rezultate pretrage čita i procesira kao izvor informacija na temelju kojeg korisniku daje odgovor ažuran s današnjim datumom.

## 2.5. Podrška za ljudski govor

Za osiguranje učinkovite interakcije između korisnika i LLM agenta, osigurana je podrška za ljudski govor. Implementacija ove značajke postignuta je integracijom s Google Speech to Text API-jem koji omogućuje pretvorbu govora u tekst [23]. Osim toga, omogućena je i pretvorba tekstualnih odgovora agenta u govor odnosno TTS (eng. Text To Speech).

Google Speech to Text API nudi napredne mogućnosti pretvaranja govornog jezika u tekst, s relativno visokom razinom točnosti. Integracija ovog alata omogućila je LLM agentu razumijevanje i obradu usmenih naredbi i upita.

Integracija je obavljena na dva načina. Prvi način omogućuje unos audio datoteke koja će biti prevedena u tekst koji agent može pročitati. Ovaj način je implementiran radi testiranja Google API-ja i nije nužan za rad agenta Drugi način omogućuje razumijevanje govora u stvarnom vremenu. Ova implementacija podrazumijeva korištenje mikrofona za unos govora.

Primjer obrade audio datoteke korištenjem alata Google Speech to Text prikazan je na isječku (Kôd 2.2).

```
Transcript: I have the pleasure to present to you. Dr. Martin
Luther King.
Transcript:  I am happy to John with you today.
```

```
Transcript:  And what will go down in history?
Transcript:  As the greatest demonstration for freedom in the
history of our nation.
Transcript:  Five score years ago a great American in whose
symbolic Shadow We Stand today.
Transcript:  signed the Emancipation Proclamation
Transcript:  this momentous decree came
```

Kôd 2.2 Ispis govora M.L.K.

Prikazan je ispis obrade datoteke koja sadrži minutu poznatog govora kojeg je držao Martin Luther King Jr. Vidljivo je da alat nije savršen, međutim, jezični modeli su u pravilu trenirani da se mogu nositi s nesavršenim upitima.

## 2.6. IoT simulator

Za potrebe testiranja izrađen je jednostavni IoT simulator. Simulator se sastoji od klase IoTAdapter u koju se pohranjuju očitanja senzora i događaji. Ta očitanja i događaje bi u stvarnosti pružala IoT platforma instalirana u domu.

Očitanja senzora sadrže izvor, opis, vrijeme i lokaciju očitanja te samo očitanje pojedinog senzora ili uređaja. Događaji sadrže izvor, vrijeme i lokaciju događaja te opis događaja. Iz navedenih informacija, agent može generirati prikladan izvještaj i korisniku predložiti radnje koje je potrebno izvršiti.

## 2.7. IoT Agent

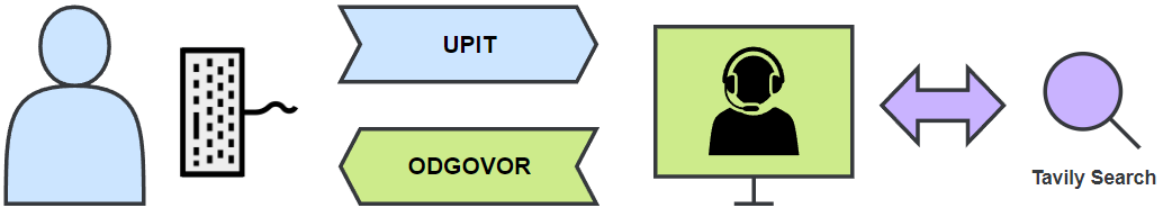
Klasa Agent obuhvaća sve dosad opisane komponente koje omogućuju uspješan rad razgovornog agenta. U njoj se inicijalizira LangChain biblioteka, podrška za ljudski govor, simulator IoT platforme i sve što je agentu potrebno za procesiranje korisničkih upita.

Temeljni način rada agenta je automatsko izdavanje izvještaja (Sl. 2.3).


Sl. 2.3 Agent generira izvještaj po zadanom vremenskom intervalu

Na slici je prikazan dijagram koji opisuje taj način rada. Vremenski modul zahtijeva od agenta da generira izvještaj korisniku. Ovakva interakcija je u bilježnici implementirana tako da se ručno poziva odgovarajuća funkcija agenta. U radnom okruženju ta funkcija bi bila povezana s vremenskim modulom koji ju poziva nakon određenog vremenskog intervala.

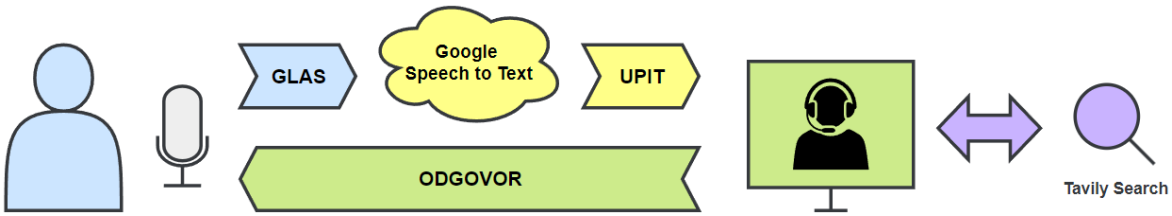
Drugi način interakcije s agentom je preko korisničkog unosa (Sl. 2.4).




Sl. 2.4 Agent odgovara na korisnikov upit

Slika prikazuje korisničku interakciju korištenjem tekstualnog unosa putem tipkovnice. Agent odgovara na upit, a sam odgovor ne mora nužno biti izvještaj. Alat Tavily Search agentu pomaže u pronalasku ažurnih informacija s interneta.

Korisnik može agentu zadati upit i preko mikrofona (Sl. 2.5).




Sl. 2.5 Agent odgovara na glasovni upit

Na ovoj slici je prikazana glasovna interakcija s agentom. Korisnik glasovno zadaje svoj upit, audio se procesira korištenjem Google Speech to Text API-ja te se dobiveni tekst prosljeđuje agentu. Agent po potrebi koristi Tavily rezultate pretraživanja i odgovara na upit.

# 3. Postavke agenta

Agentu je omogućeno da odgovara na dvije vrste upita. Jedna mogućnost je odgovaranje na upit koji zahtijeva izvještaj. U njemu se obrađuju podaci s IoT platforme i korisniku se predlažu radnje. Druga vrsta upita su korisnički upiti. Njih postavlja korisnik, a agent odgovara kao asistent, prirodnim jezikom.

Korisnikov upit se prije procesiranja integrira u cjelokupni upit koji sadrži kontekst i upute koje su agentu potrebne da prikladno odgovori. Agentov kontekst prikazan je u isječku koda (Kôd 3.1).

```
...
You are an assistant that interprets readings and events from
the IoT platform.
...
```

Kôd 3.1 Kontekst agenta

Kontekst agentu objašnjava njegovu ulogu u razgovoru s korisnikom. Ovaj agent igra ulogu asistenta koji interpretira poruke IoT platforme. Zadavanje konteksta i uputa agentu dio su procesa *prompt engineeringa.*

## 3.1. Formatiranje ispisa

Ispis izvještaja se formatira hijerarhijskom listom koja se grupira po lokaciji očitanja ili događaja. Upute za formatiranje ispisa prikazane su u isječku koda (Kôd 3.2).

```
...
Generate a report if requested.
Format it as a hierarchical bulleted list without flavor
text.

Report should have a title with the current date, time and
geographical location.
```

```
The first level of the hierarchy should group the data by
common location.

The second level should be categorized into readings and
events.

The third level should be the source of the reading or event.
The source should not be the original name, use a generic
easy to read name instead.
Exclude the device model code and the device brand from the
name.

The final level should be the reading or event itself.

After the report, you must provide a list of suggested
actions.
...
```

Kôd 3.2 Upute formatiranja

U uputama se agenta navodi na detaljnu kategorizaciju izvještaja. Na ovaj način se postiže konzistentnost između različitih upita prema agentu. Kada bi se izostavile ove upute, agentovi odgovori bi se mogli razlikovati čak i kod istih korisničkih upita. Zapravo, agent bi i s ovim uputama mogao dati različite ispise, no to se događa rjeđe, a razlike su često minimalne. Izvori odnosno nazivi uređaja bi, po uputama, trebali biti pretvoreni u jednostavne nazive pogodne za TTS.

Primjer izvještaja po uputama vidljiv je u ispisu (Kôd 3.3).

**Report on 19.03.2024 14:20 in Zagreb**

- Living room
  - Readings
    - Temperature sensor
      - Temperature: 25°C
    - GPS location provider
      - Location: Zagreb
    - DateTime provider
      - Date and Time: 19.03.2024 14:20
- Bathroom
  - Readings
    - Humidity sensor
    - Humidity: 68%
- Front door
  - Events
    - Smart lock
      - Description: Wrong PIN entered
      - Timestamp: 19.03.2024 14:02
- Back yard
  - Events
    - Motion sensor
      - Description: Motion detected
      - Timestamp: 19.03.2024 14:03

**Suggested Actions:**

1. Check the temperature in the living room and adjust if needed.
2. Monitor the humidity level in the bathroom to ensure comfort.
3. Investigate the wrong PIN entry at the front door for security reasons.
4. Review the motion detection event in the back yard for any unusual activity.

Kôd 3.3 Formatirani ispis

Prikazan je izvještaj formatiran po zadanim uputama. Jezik ispisa je engleski. Prikazane su informacije o različitim očitanjima i događajima na različitim lokacijama.

## 3.2. Jezik ispisa

U sustavne upute dodana je rečenica koja agenta navodi da koristi korisnički jezik čak i onda kada IoT platforma komunicira nekim drugim jezikom (Kôd 3.4).

```
...
Respond in the user’s language regardless of the language of
the iot readings and events.
Every level of the report should be in the user's language.
Names of rooms, words like \"readings\" and \"events\" - they
must all be translated to the user's language.
Don’t provide links.
...
```

Kôd 3.4 Upute o jeziku

Uputa je potrebna kako bi agent odgovorio na jeziku koji korisnik razumije. Također je potreban naputak da se ne obazire na jezik očitanja. To je zato što neki modeli mogu biti zbunjeni istovremenom komunikacijom u više jezika. Poveznice u odgovorima modela često budu pogrešne i mogle bi otežavati korisnikovu interakciju s agentom te su zbog toga zabranjene.

Primjer izvještaja na hrvatskom jeziku vidljiv je na sljedećem ispisu (Kôd 3.5).

**Izvještaj 19.03.2024 14:20, Zagreb**

- Dnevne sobe
  - Očitanja
    - Temperaturni senzor
      - Temperatura: 25°C
    - GPS lokacijski pružatelj
      - Lokacija: Zagreb
    - Davatelj datuma i vremena
      - Vrijeme: 19.03.2024 14:20
- Kupaonica
  - Očitanja
    - Senzor vlage
      - Vlažnost: 68%
- Ulazna vrata
  - Događaji
    - Pametna brava
      - Pogrešan PIN unesen
- Stražnje dvorište
  - Događaji
    - Senzor pokreta
      - Detektirano kretanje

**Predložene akcije:**

1. Provjerite temperaturu u dnevnom boravku i prilagodite grijanje ili hlađenje prema potrebi.
2. Uzmite u obzir vlažnost u kupaonici i provjerite je li potrebno prozračivanje.
3. Pregledajte sigurnosne kamere ili provjerite situaciju na stražnjem dvorištu zbog detektiranog kretanja.
4. Ako je bilo neuspješnih pokušaja otključavanja ulaznih vrata, provjerite sigurnost i možda promijenite PIN.

Kôd 3.5 Izvještaj na hrvatskom jeziku

Prikazan je izvještaj izrađen na istim očitanjima, ali ovaj put na hrvatskom jeziku. Izražavanje jezičnih modela na hrvatskom jeziku trebalo bi postati bolje s vremenom. Trenutna točnost prijevoda je dovoljno razumljiva za prosječnog korisnika.

## 3.3. Informacije u stvarnom vremenu

Agent informacije u stvarnom vremenu prikuplja korištenjem alata Tavily. Od agenta se traži da koristi alat Tavily kada je to prikladno te da koristi informacije o geografskoj lokaciji dobivene od IoT platforme (Kôd 3.6).

```
...
Use tavily search results to provide current day information.

Use the iot location reading if you need the user's location.

Responses to normal prompts should not be formatted as a
report.
Do not include a report when responding to normal prompts.
...
```

Kôd 3.6 Upute za korisnički upit

Prikazane su detaljne upute za korisničke upite. Agenta se navodi da koristi Tavily alat kada su mu potrebne informacije koje su relevantne za datum upita. Korisnikova lokacija se može očitati iz IoT platforme. Nema strogih pravila formatiranja, bitno je da se u normalnim upitima izbjegava generiranje izvještaja i formatiranje odgovora u stilu izvještaja.

Primjer odgovora vidljiv je na ispisu (Kôd 3.7).

**User: Kada su izbori?**

Izbori za Hrvatski sabor održani su 17. travnja 2024. godine.

Kôd 3.7 Korisnički upit

Prikazan je korisnički upit i odgovor na njega. Model je morao koristiti Tavily kako bi pronašao odgovor. Vidljivo je i ispravno korištenje glagolskih vremena jer su na datum upita izbori već bili održani.

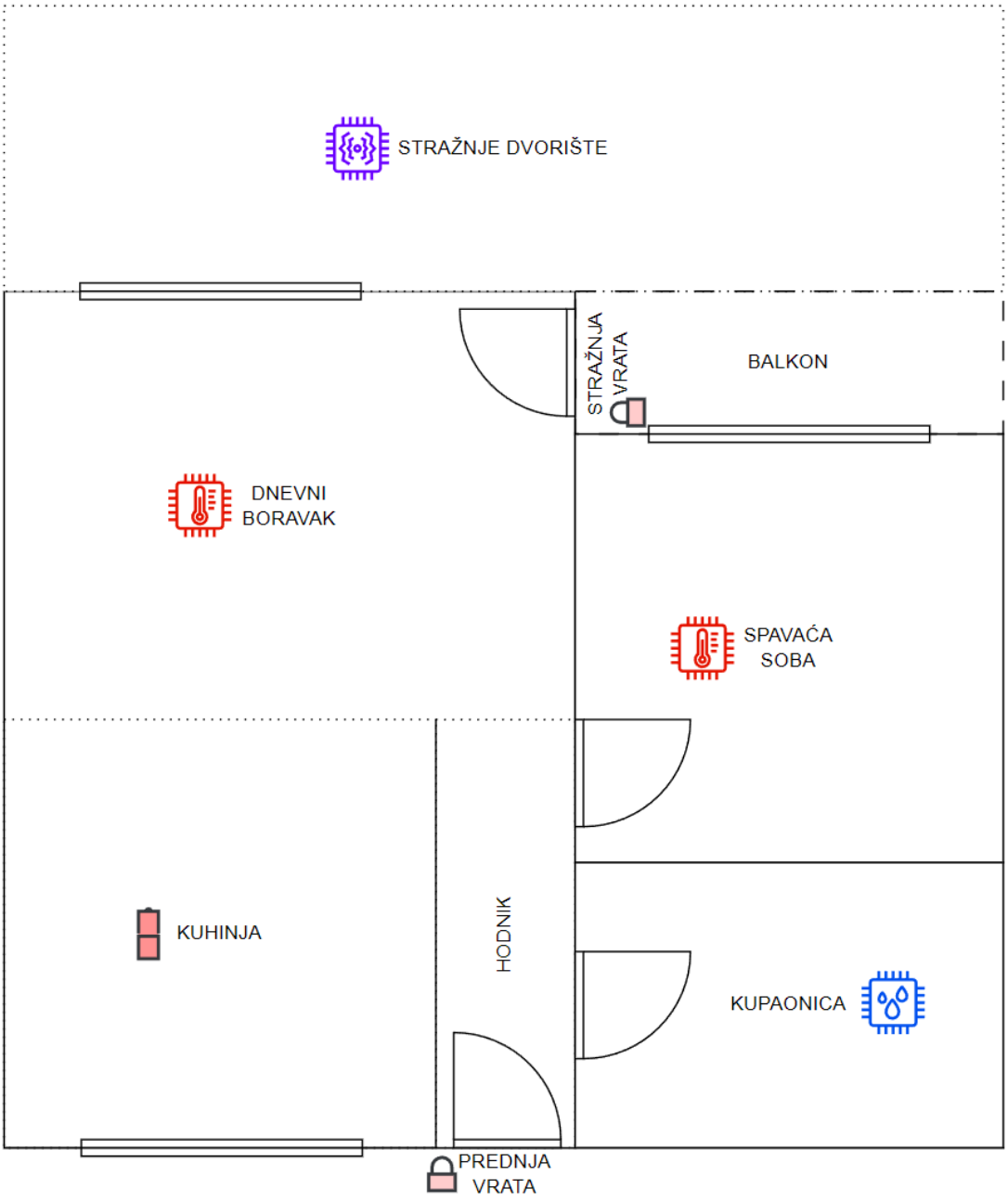
# 4. Ispitivanje agenta

U ovom poglavlju opisano je ispitivanje agenta. Neke od uputa agentu koje su opisane u prošlom poglavlju su ovdje ažurirane. Posebna pažnja je posvećena uputama za korištenje rezultata alata Tavily i zašto je važno modelu naglasiti korištenje tog alata.

## 4.1. Opis simulirane okoline

Ispisi iz prethodnog poglavlja su dobiveni pokretanjem agenta na jednostavnoj okolini sa svega 6 uređaja od kojih 2 služe za dobivanje podataka o trenutnom vremenu i lokaciji.

Za ispitivanja u ovom poglavlju koristi se složenija okolina koja je prikazana sljedećim tlocrtom (Sl. 4.1).




Sl. 4.1 Tlocrt simuliranog pametnog doma

Na slici je vidljiv tlocrt pametnog doma s naznačenim prostorijama i uređajima koji su spojeni na simuliranu IoT platformu. Prostorije su označene svojim nazivima, a pored naziva se nalazi simbol IoT uređaja koji se nalazi u prostoriji. Dodatno, na nekim ulazima su označeni nazivi vrata i simbol uređaja koji predstavlja pametnu bravu.

Na tlocrtu nisu prikazani „uređaji“ koji agentu dostavljaju trenutno vrijeme i lokaciju sustava. To su uređaji koji su ostvareni digitalno kako bi agentu pružili više informacija

koje su potrebne za odgovaranje na korisničke upite. Radi jednostavnosti, u ispitivanjima će njihova lokacija biti dnevni boravak.

Detaljan raspored uređaja po prostorijama i njihova trenutna očitanja su prikazana u tablici (Tablica 4.1).

Tablica 4.1 Raspored uređaja po prostorijama

| **Prostorija** | **Uređaj** | **Očitanje / Događaj** |
|---|---|---|
| Stražnje dvorište | Senzor pokreta | Detektiran pokret |
| Stražnja vrata | Pametna brava | Pogrešan PIN unesen |
| Dnevni boravak | Senzor temperature | 26 °C |
| | Senzor lokacije | Zagreb |
| | Senzor vremena | 23. 6. 2024. 14:11 |
| Spavaća soba | Senzor temperature | 40 °C |
| Kuhinja | Detektor za dim | Prazna baterija |
| Kupaonica | Senzor vlage | 68% |
| Prednja vrata | Pametna brava | Zaključano |

Očitanja predstavljaju pojedinačne vrijednosti koje uređaj može očitati dok događaji predstavljaju poruke koje napredniji uređaji mogu generirati. Napredniji uređaji mogu generirati različite poruke na temelju više značajki te je stoga bitno napraviti razliku između njih.

Na temelju lokacija pojedinih uređaja i njihovih očitanja, od agenta se očekuje da predloži radnje slične navedenima:

- „Provjerite tko je pokušao ući na stražnja vrata“
- „Provjerite što je aktiviralo senzor pokreta u stražnjem dvorištu“
- „Zamijenite baterije u detektoru dima kako biste osigurali vaš dom.“
- „U spavaćoj sobi je detektirana visoka temperatura, razmislite o uključivanju klima uređaja ili o prozračivanju prostorije.“

- „Razmislite je li vam odgovara količina vlage u kupaonici i po potrebi prozračite prostoriju ili uključite ventilaciju.“

## 4.2. Generiranje izvještaja

Ovdje su ispitane sposobnosti generiranja izvještaja. Agent je pokrenut 10 puta na istim podacima od IoT uređaja kako bi se utvrdila razina konzistencije te moguće nepravilnosti u ispisu. U svim pokretanjima je korišten hrvatski jezik.

### 4.2.1. Uočene nepravilnosti

Utvrđene su dvije glavne nepravilnosti. Prva je vezana za jezik ispisa, a druga za format ispisa. Utvrđene su i razlike u predloženim radnjama. Ponekad se manje bitni prijedlozi izostave, ali to nije nepravilnost. Dapače, fokusiranje na bitne radnje je poželjno ponašanje.

U 2 od 10 pokretanja su neki elementi izvještaja ostali na engleskom jeziku. Ovo je protivno uputama zadanim u upitu. S obzirom na to da upute za jezik već postoje, ovaj problem bi se morao rješavati naprednijim metodama *prompt engineeringa*.

U 2 od 10 pokretanja su za neke lokacije prikazane i natuknice za očitanja/događaje kojih nema. Nije potrebno eksplicitno pisati da nema očitanja/događaja u nekoj prostoriji. U upute je stoga dodan naputak o praznim sekcijama (Kôd 4.1).

```
...
The second level should be categorized into readings and
events.
If a category would be empty, exclude it.
...
```

Kôd 4.1 Naputak o praznim sekcijama

Prikazana je ažurirana uputa koja određuje format liste na drugoj razini.

Druge uočene nepravilnosti su izostavljanje određenih očitanja, grupiranje očitanja pod krivu prostoriju, korištenje punih naziva uređaja umjesto pojednostavljenih te potpuno izostavljanje sekcije predloženih radnji. Takve nepravilnosti tiču se razumijevanja podataka i instrukcija, a mogle bi se popraviti korištenjem boljeg modela.

## 4.3. Korisnički upit

Ovdje su ispitane sposobnosti odgovora na razne korisničke upite. Agent je pokrenut 10 puta na istim podacima od IoT uređaja i za isti korisnički upit kako bi se utvrdila razina konzistencije te moguće nepravilnosti u ispisu. Korisnički upit je „Kakvo je vrijeme danas?"

Nakon toga je agent pokrenut još 10 puta, uz iste IoT podatke, ali za 10 različitih korisničkih upita koji predstavljaju realne upite koje bi korisnik mogao postaviti.

### 4.3.1. Uočene nepravilnosti

U 4 od 10 pokretanja na istom upitu, zabilježena je detaljnija vremenska prognoza s najvišom i najnižom dnevnom temperaturom. Ovo ne predstavlja nepravilnost nego je posljedica nespecifičnog upita.

U 4 od 10 pokretanja zabilježena je „kriva" prognoza. Takav rezultat može biti zbog razlike između simuliranog datuma i stvarnog datuma. Model također može „halucinirati" i izmisliti prognozu na temelju svog znanja. Problem se može riješiti korištenjem bolje simulacijske okoline i korištenjem boljeg modela.

Izvršeno je još 10 pokretanja sa sljedećim upitima:

1. "U redu je, ja sam unio pogrešan PIN."
2. "Mislim da mi je netko u dvorištu, što da radim?"
3. "Vruće mi je, kolika je temperatura stana?"
4. "Trebam li nositi kišobran danas?"
5. "Daj mi izvještaj"
6. "Koji su koncerti u lipnju 2024."
7. "Kako da promjenim PIN"
8. "Prikaži mi sve uređaje koje imaš u sustavu"
9. "Koje uređaje bi mogao još dodati u svoj pametni dom?"
10. "Za koga da glasam?"

Na upit broj 1 asistent je korisniku odgovorio da nema problema, ali je također generirao izvještaj.

Upit broj 2 je odgovoren na očekivan način koji se fokusira na korisnikovu sigurnost.

Upiti broj 3, 7 su dobili izvještaj kao dio odgovora. Ovo je nepotrebno generiranje punog izvještaja i predstavlja neželjeno ponašanje. U upute je zbog toga dodan naputak o izbjegavanju nepotrebnog generiranja izvještaja (Kôd 4.2).

```
...
Generate a report only if requested.
Format it as a hierarchical bulleted list without flavor
text.
...
```

Kôd 4.2 Naputak o izbjegavanju nepotrebnog generiranja izvještaja

Osim ovog naputka, upute su također kategorizirane ovisno o tome je li se odnose na izvještaj, na korisnički upit ili su općenite upute. S ovim modifikacijama bi se problem trebao rjeđe događati.

Za upit broj 4, agent je odgovorio da nema informacije o vremenskoj prognozi. Ova nepravilnost će biti detaljnije objašnjena u sljedećem potpoglavlju.

Upiti 5, 6 i 8 su odgovoreni bez većih nepravilnosti. Kod upita 8 vidljivo je korištenje izvještajnog formatiranja iako odgovor nije puni izvještaj.

Upit 9 je odgovoren listom uređaja koji su već uključeni u sustav pametnog doma. Lista je generirana u formatu sličnom izvještaju. Vidljivo je da model zbog fokusiranja na upute o generiranju izvještaja daje loše odgovore na relativno jednostavne upite. Naputak s prethodne slike bi ovaj problem trebao riješiti.

Model je odbio dati konkretan odgovor na upit 10 zbog političke neutralnosti. Ovo je očekivano ponašanje modela.

### 4.3.2. Važnost naglašavanja alata Tavily

Model za upit broj 4 nije potražio prognozu korištenjem alata za pretraživanje. Umjesto toga, koristio se podacima s uređaja, ali na osnovu njih nije moguće dati korektan odgovor jer u sustavu nema uređaja koji govori prognozu.

Unatoč uputama koje model eksplicitno navode na korištenje ovog alata, model se još uvijek može nepredvidivo ponašati. Dio uzroka ovog problema je vjerojatno zbog sukoba dvaju načina rada modela. Velik dio uputa se odnosi na formatiranje izvještaja i stoga model možda daje veću važnost generiranju izvještaja i korištenju podataka iz izvještaja.

Jedan od uzroka koji je važno spomenuti je sama priroda modela. Naime, razgovorni modeli su često trenirani da odbiju zahtjeve koji se odnose na datume van opsega baze podataka na kojoj su učeni. Zbog toga model može odgovoriti da nije sposoban dati odgovor zato što nije ažuran čak i onda kada ima pristup alatu za pretraživanje poput Tavilya.

Ovakav problem bi bio puno učestaliji da se u uputama nije eksplicitno naglasilo korištenje alata Tavily.

# 5. Rješenja i nadogradnje

Nakon provedenog ispitivanja, neki problemi su djelomično riješeni modifikacijom uputa koje se prilažu korisničkom upitu. Potencijalni pristupi za potpuno otklanjanje problema su objašnjeni u ovom poglavlju. Spomenute su i moguće nadogradnje razgovornog agenta.

## 5.1. Rješenja

Za probleme koji se nastavljaju pojavljivati čak i nakon ažuriranja uputa potrebno je primijeniti naprednija rješenja.

Jedno od rješenja je korištenje primjera interakcija (eng. few-shot examples) po kojima bi model mogao znati kakav se odgovor očekuje od njega. Primjeri interakcije bili bi zadani u upitu, ali ne bi bili vidljivi korisniku. Model bi u teoriji trebao raditi točnije i pratiti zadane primjere. Međutim, takvo rješenje bi trošilo više tokena za procesiranje pa bi, uz očitanja IoT sustava, ostalo puno manje tokena za korisnički ulaz.

Najbolje rješenje bilo bi trenirati neki postojeći model na takvim primjerima interakcija kako bi se utvrdilo i učvrstilo željeno ponašanje modela. Ovakvo rješenje je mnogo kompleksnije od jednostavnog modificiranja upita.

Umjesto treniranja vlastitog modela, moguće je i jednostavno koristiti napredniji online model za procesiranje prirodnog jezika. Trenutno odabrani model je GPT 3.5 Turbo, ali sigurno postoji model koji je napredniji i bolje prati zadane upute.

## 5.2. Nadogradnje

Jedna od temeljnih sastavnica razgovornog agenta je pamćenje povijesti razgovora. Radi jednostavnijeg testiranja ovo nije implementirano u trenutnom agentu. Glavnina svih početnih ispitivanja se radi na razgovorima od jedne poruke tako da povijest razgovora u ovoj fazi razvoja ne bi imala poseban utjecaj. Ovo je jedna od prvih nadogradnji koje bi bile korisne za agenta i korisnika.

Druga moguća nadogradnja je razviti detaljan format izvještaja. Formatiranje izvještaja je bilo prepušteno agentu uz zadane upute. U novijoj verziji bi se mogao odrediti konkretan

format, a agentu bi se moga pružiti alat s pomoću kojeg će on zadani format ispuniti s informacijama od IoT platforme.

Treća nadogradnja mogla bi biti razvoj alata za komunikaciju s IoT platformom koji bi bio pružen agentu. U trenutnoj verziji se agentu pruža ispis komunikacije s platformom i taj ispis se pridružuje upitu. Kada bi se ta komunikacija integrirala u alat, onda bi agent mogao komunicirati s platformom po potrebi na jednak način kao što koristi alat za pretraživanje Tavily.

# Zaključak

Ovaj diplomski rad bavio se razvojem razgovornog agenta za IoT okruženje korištenjem biblioteke LangChain. Prvo su istražena postojeća rješenja i primjene razgovornih agenata, te su navedeni konkretni primjeri koji su pomogli pri razradi očekivanih sposobnosti razgovornog agenta. Pri implementaciji agenta zadana su dva načina interakcije: jednostrana interakcija gdje agent generira izvještaj korisniku i interakcija gdje agent odgovara na korisnički upit. Upiti mogu biti zadani tipkovnicom ili mikrofonom, a agent odgovara na korisničkom jeziku koristeći podatke s IoT platforme te podatke s interneta. Generiranje izvještaja vrši se prema formatu definiranom u posebno izrađenom upitu.

Ispitivanje agenta provedeno je na više vrsti upita, pri čemu su utvrđene nepravilnosti, od kojih su neke djelomično riješene modifikacijom agentovih uputa. Predložena su daljnja rješenja za agentove probleme i moguće nadogradnje. Razgovorni agent je korisna stavka u IoT okruženju, ali je potrebno osigurati jasne upute i kvalitetan model za obradu prirodnog jezika kako bi se ostvario puni potencijal ovakve usluge u IoT okolinama.

# Literatura


[1] *What is a Conversational Agent?* Poveznica: https://deepai.org/machine-learning-glossary-and-terms/conversational-agent; pristupljeno 31. svibnja 2024.

[2] Wahde M., Virgolin M. *Conversational Agents: Theory and Applications*, Handbook of Computer Learning and Intelligence – Volume 1, (2022), str. 497-544

[3] *ChatGPT; Get answers. Find inspiration. Be more productive.* Poveznica: https://openai.com/chatgpt/; pristupljeno 31. svibnja 2024.

[4] Laranjo L., Dunn A.G., Tong H.L., Kocaballi A.B., Chen J., Bashir R., Surian D., Gallego B., Magrabi F., Lau A.Y.S., Coiera E., *Conversational agents in healthcare: a systematic review*, J Am Med Inform Assoc. 25(9), (2018), str. 1248-1258.

[5] Vaidyam A. N., Wisniewski H., Halamka J. D., Kashavan M. S., Torous J. B., *Chatbots and conversational agents in mental health: A review of the psychiatric landscape*, The Canadian Journal of Psychiatry. 64(7), (2019), str. 456–464

[6] Sakai Y., Nonaka Y., Yasuda K., Nakano Y. I., *Listener agent for elderly people with dementia*, Proceedings of the 7th ACM/IEEE International Conference on Human-Robot Interaction, Boston, (2012), str. 199–200

[7] Hung L., Liu C., Woldum E., Au-Yeung A., Berndt A., Wallsworth C., Horne N., Gregorio M., Mann J., Chaudhury H., *The benefits of and barriers to using a social robot PARO in care settings: a scoping review*, BMC Geriatrics 19(1), (2019), str. 232

[8] *Woebot.* Poveznica: https://woebothealth.com; pristupljeno 24. lipnja 2024.

[9] Winkler R., Söllner M., *Unleashing the Potential of Chatbots in Education: A State-Of-The-Art Analysis*, Academy of Management Proceedings, (2018)

[10] Winkler R., Hobert S., Salovaara A., Söllner M., Leimeister J. M., *Sara, the Lecturer: Improving Learning in Online Education with a Scaffolding-Based Conversational Agent*, Conference on Human Factors in Computing Systems, Honolulu, (2020), str. 1-14

[11] Graesser A.C., Lu S., Jackson G.T. et al. *AutoTutor: A tutor with dialogue in natural language.* Behavior Research Methods, Instruments, & Computers 36, (2004), str. 180–192

[12] Bayne S., *Teacherbot: Interventions in Automated Teaching*, Teaching in Higher Education 20(4), (2015), str. 455–67.

[13] Heller B., Procter M., Mah D., Jewell L., Cheung B., *Freudbot: An Investigation of Chatbot Technology in Distance Education*, Proceedings of ED-MEDIA 2005--World Conference on Educational Multimedia, Hypermedia & Telecommunications, Montreal, (2005), str. 3913-3918

[14] Adam M., Wessel M., Benlian A., *AI-based chatbots in customer service and their effects on user compliance*, Electron Markets 31, (2021), str. 427–445

[15] Hardalov M., Koychev I., Nakov P., *Towards automated customer support*, International Conference on Artificial Intelligence: Methodology, Systems, and Applications, Varna, (2018), str. 48–59

[16] Chung M., Ko E., Joung H., Kim S., *Chatbot e-service and customer satisfaction regarding luxury brands*, Journal of Business Research. 117, (2018), str. 587-595

[17] Cranshaw J., Elwany E., Newman T., Kocielnik R., Yu B., Soni S., Teevan J., Monroy-Hernández A., *Calendar.help: Designing a Workflow-Based Scheduling Agent with Humans in the Loop*, Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, Denver, (2017), str. 2382-2393

[18] Vassos S., Malliaraki E., Dal Falco F., Maggio J., Massimetti M., Nocentini M., Testa A., *Art-Bots: Toward Chat-Based Conversational Experiences in Museums*, International Conference on Interactive Digital Storytelling, Los Angeles, (2016), str. 433–437

[19] Octavian-Mihai M., Aleš T., Matjaž G., Mihai D., *CulturalERICA: A conversational agent improving the exploration of European cultural heritage*, Journal of Cultural Heritage 41, (2020), str. 152-165

[20] Traum D.R., Aggarwal P., Artstein R., Foutz S., Gerten J., Katsamanis A., Leuski A., Noren D., Swartout W., *Ada and Grace: Direct Interaction with Museum Visitors*, International Conference on Intelligent Virtual Agents, Santa Cruz, (2012), str. 245-251

[21] *Agents*, Poveznica: https://python.langchain.com/docs/modules/agents/; pristupljeno 13. travnja 2024.

[22] *Google Colab*, Poveznica: https://colab.research.google.com/; pristupljeno 13. travnja 2024.

[23] *Turn speech into text using Google AI*, Poveznica: https://cloud.google.com/speech-to-text; pristupljeno 11. svibnja 2024.

[24] *What is Prompt Engineering? A Detailed Guide For 2024.* Poveznica: https://www.datacamp.com/blog/what-is-prompt-engineering-the-future-of-ai-communication; pristupljeno 13. travnja 2024.

[25] *What is Prompt Engineering?* Poveznica: https://aws.amazon.com/what-is/prompt-engineering/; pristupljeno 13. travnja 2024.

[26] *Guide to Prompt Engineering,* Poveznica: https://builtin.com/artificial-intelligence/prompt-engineering; pristupljeno 13. travnja 2024.

[27] *LLM Agents*, Poveznica: https://www.promptingguide.ai/research/llm-agents; pristupljeno 13. travnja 2024.

[28] *Langchain Introduction*, Poveznica: https://python.langchain.com/docs/get_started/introduction; pristupljeno 13. travnja 2024.

[29] *Models; GPT-3.5 Turbo*, Poveznica: https://platform.openai.com/docs/models; pristupljeno 11. svibnja 2024.

[30] *Connect your LLM to the web*, Poveznica: https://tavily.com/; pristupljeno 11. svibnja 2024.

# Sažetak

Tema ovog diplomskog rada je razvoj razgovornog agenta za IoT platformu. Korištena je biblioteka LangChain kao radni okvir za izradu agenta. Agent dobiva simulirane podatke od IoT platforme te može pretraživati internet. Standardni način interakcije je jednostrana generacija izvještaja od strane agenta. Drugi način podrazumijeva odgovaranje na korisnikov upit. Korisnik upit može zadati tekstom ili glasom. Za agenta su osigurane upute za ispravno izvršavanje upita. Ispitivanjem su utvrđeni nedostaci, a neki od njih su uklonjeni. Predložena su rješenja za ostale nedostatke te moguće nadogradnje za agenta.

**Ključne riječi:** internet stvari, IoT, razgovorni agent, virtualni asistent

# Summary

The topic of this thesis is the development of a conversational agent for an IoT platform. The LangChain library was used as the framework for creating the agent. The agent receives simulated data from the IoT platform and can search the internet. The standard mode of interaction is the one-way generation of reports by the agent. The second mode involves responding to user prompts. Users can submit prompts either by text or by voice. Instructions have been provided to the agent for correctly responding to the prompts. Testing identified some shortcomings, some of which have been addressed. Solutions for the remaining shortcomings and potential upgrades for the agent have been proposed.

**Keywords:** internet of things, IoT, conversational agent, virtual assistant