Ime, priimek: Žiga Fon

LV03: Pogovorni sistem Amazon Alexa

1.1 Poročilo o preizkusih vaše Aleksa veščine

Opis veščine, glasovni vmesnik (Klicna fraza (skill invocation), definicije vaših fraz, intentov, slotov,.. ali screenshoti)

Ime storitve: RENISHAW INFO

Klicna fraza: open application specific integrated circuit company

Welcome to slovenia Renishaw company info. You can say employees, products. Which one would you like to try?

Intent : Namen	Fraze (Sample Utterances)	Vrednosti parametrov (slot)	Kaj bo odgovor
HelloWorldIntent	hello how are you	No parameters	Welcome to Renishaw company info page.
FactsIntent	Give me {info}	Employees	This is a fact about employees. Renishaw has offices and employees and representatives in over 36 countries worldwide
FactsIntent	Give me {info}	Products	This is a fact about products. Renishaw's products are widely used in aerospace, automotive, electronics, and healthcare industries.The company pioneered the touch-trigger probe, which transformed manufacturing measurement.Renishaw's additive manufacturing solutions involve high-precision metal 3D printing.
FactsIntent	I want {info}	Products	This is a fact about products. Renishaw's products are widely used in aerospace, automotive, electronics, and healthcare industries.The company pioneered the touch-trigger probe, which transformed manufacturing measurement.Renishaw's additive

			manufacturing solutions involve high-precision metal 3D printing.
FactsIntent	I want {info}	Employees	This is a fact about employees. Renishaw has offices and employees and representatives in over 36 countries worldwide

Model interakcije v JSON obliki : (Intents – JSON editor)

Vse lahko izvodzimo kot model pogovora v JSON obliki.

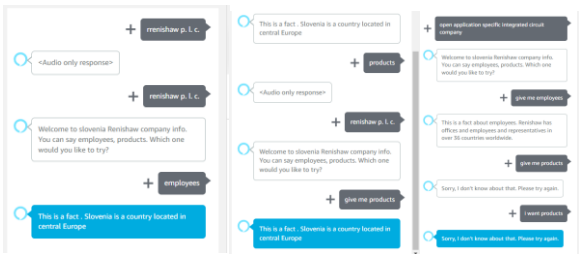

```
{
  "interactionModel": {
    "languageModel": {
      "invocationName": "application specific integrated circuit company",
      "intents": [
        {
          "name": "AMAZON.CancelIntent",
          "samples": []
        },
        {
          "name": "AMAZON.HelpIntent",
          "samples": []
        },
        {
          "name": "AMAZON.StopIntent",
          "samples": []
        },
        {
          "name": "HelloWorldIntent",
          "slots": [],
          "samples": [
            "hello",
            "how are you",
            "say hi world",
            "say hi",
            "hi",
            "say hello world",
            "say hello"
          ]
        },
        {
          "name": "AMAZON.NavigateHomeIntent",
          "samples": []
        },
        {
          "name": "AMAZON.FallbackIntent",
          "samples": []
        },
        {
          "name": "FactsIntent",
          "slots": [
            {
              "name": "info",
              "type": "Info"
            }
```

```
            ],
            "samples": [
              "i want {info}",
              "give me {info}",
              "information",
              "info"
            ]
          }
        ],
        "types": [
          {
            "name": "Info",
            "values": [
              {
                "name": {
                  "value": "products"
                }
              },
              {
                "name": {
                  "value": "employees"
                }
              }
            ]
          }
        ]
      }
    }
}
```


Način generiranja odgovora (delovanje kode, prilagoditve glede na vzorec, dodatne funkcije v kodi..)

Koda je pa v ozadju javaskript, s katero lahko sprogramiramo amazon Alekso, da poveže vprašanja z odgovori. Spremenil sem tematiko odgovorov, ne gre več za turizem temvel za aplikacijoza deljenje informacij o podjetju

Testiranje storitve (screenshot poteka pogovora)



Komentar po testiranju : uporabnost, zanesljivost delovanja (npr. ali pravilno zazna odpiranje vaše storitve, ali prepozna namen (intent) ter fraze..), prijaznost. Razlika med testiranjem v razvojnem okolju, ter testiranjem na napravi (Amazon Echo).

Če ne poveš točno tisto kar želiš ti Aleksa odda en nov odgovor iz svojih strežnikov ki ga nisi sam sprogramiral. Podatke lahko kar povleče iz Wikipedije in itd. Z Aleksa stop jo utišamo.

Kompletna koda (lambda funkcija, index.js) :

```
const Alexa = require('ask-sdk-core');

const LaunchRequestHandler = {
  canHandle(handlerInput) {
    return Alexa.getRequestType(handlerInput.requestEnvelope) === 'LaunchRequest';
  },
  handle(handlerInput) {

    const speakOutput = 'Welcome to slovenia Renishaw company info. You can say employees, products. Which one would you like to try?';
    const repromptOutput = 'You can say employees, products. Which one would you like to try?';

    return handlerInput.responseBuilder
      .speak(speakOutput)
      .reprompt(repromptOutput)
      .getResponse();
  }
};

const HelloWorldIntentHandler = {
  canHandle(handlerInput) {
    return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest'
      && Alexa.getIntentName(handlerInput.requestEnvelope) === 'HelloWorldIntent';
  },
  handle(handlerInput) {
    const speakOutput = 'Welcome to Renishaw company info page.';
    const repromptOutput = 'Welcome to Renishaw company info page.';

    return handlerInput.responseBuilder
      .speak(speakOutput)
      .reprompt(repromptOutput)
      .getResponse();
  }
};


const facts = [
  'Renishaw is a UK-based engineering and scientific technology company.',
  'Renishaw was founded in 1973 by Sir David McMurtry and John Deer.',
  'Renishaw specializes in precision measurement, additive manufacturing, and healthcare technology.',
  '3Renishaw's products are widely used in aerospace, automotive, electronics, and healthcare industries.',
  'Renishaw is known for its expertise in metrology and precision engineering.',
  'Renishaw has offices and employees and representatives in over 36 countries worldwide.',
  'The company pioneered the touch-trigger probe, which transformed manufacturing measurement.',
  'Renishaw's additive manufacturing solutions involve high-precision metal 3D printing.'
```

```
];

const {
 getRequestType,
 getIntentName,
 getSlotValue,
 getDialogState,
} = require('ask-sdk-core');


// core functionality for fact skill
const FactHandler = {
 canHandle(handlerInput) {
  const request = handlerInput.requestEnvelope.request;
  // checks request type
  return (request.type === 'IntentRequest'
    && request.intent.name === 'FactsIntent');
 },
 handle(handlerInput) {
  const requestAttributes = handlerInput.attributesManager.getRequestAttributes();

  const slotValue = getSlotValue(handlerInput.requestEnvelope, 'info');
  console.log('${slotValue}');

  var speakOutput = 'This is a fact '
  if (slotValue !== null) {
    speakOutput += 'about ' + slotValue;

    if (slotValue === 'employees') speakOutput += '. ' +facts[5];
    else if (slotValue === 'products') speakOutput += '. ' +facts[3]+facts[6]+facts[7] ;
    else speakOutput += facts[1];

  } else {
    speakOutput += '. ' + facts[0];
  }

  return handlerInput.responseBuilder
   .speak(speakOutput)
   // Uncomment the next line if you want to keep the session open so you can
   .reprompt('Try again')
   .getResponse();
 },
};


const HelpIntentHandler = {
```

```
  canHandle(handlerInput) {
    return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest'
      && Alexa.getIntentName(handlerInput.requestEnvelope) === 'AMAZON.HelpIntent';
  },
  handle(handlerInput) {
    const speakOutput = 'You can say hello to me! How can I help?';

    return handlerInput.responseBuilder
      .speak(speakOutput)
      .reprompt(speakOutput)
      .getResponse();
  }
};

const CancelAndStopIntentHandler = {
  canHandle(handlerInput) {
    return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest'
      && (Alexa.getIntentName(handlerInput.requestEnvelope) === 'AMAZON.CancelIntent'
        || Alexa.getIntentName(handlerInput.requestEnvelope) === 'AMAZON.StopIntent');
  },
  handle(handlerInput) {
    const speakOutput = 'Goodbye!';

    return handlerInput.responseBuilder
      .speak(speakOutput)
      .getResponse();
  }
};
/* *
 * FallbackIntent triggers when a customer says something that doesn’t map to any intents in your skill
 * It must also be defined in the language model (if the locale supports it)
 * This handler can be safely added but will be ingnored in locales that do not support it yet
 * */
const FallbackIntentHandler = {
  canHandle(handlerInput) {
    return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest'
      && Alexa.getIntentName(handlerInput.requestEnvelope) === 'AMAZON.FallbackIntent';
  },
  handle(handlerInput) {
    const speakOutput = 'Sorry, I don\'t know about that. Please try again.';

    return handlerInput.responseBuilder
      .speak(speakOutput)
      .reprompt(speakOutput)
      .getResponse();
  }
```

```
};
/* *
 * SessionEndedRequest notifies that a session was ended. This handler will be triggered when a currently open
 * session is closed for one of the following reasons: 1) The user says "exit" or "quit". 2) The user does not
 * respond or says something that does not match an intent defined in your voice model. 3) An error occurs
 * */
const SessionEndedRequestHandler = {
  canHandle(handlerInput) {
    return Alexa.getRequestType(handlerInput.requestEnvelope) === 'SessionEndedRequest';
  },
  handle(handlerInput) {
    console.log(`~~~~ Session ended: ${JSON.stringify(handlerInput.requestEnvelope)}`);
    // Any cleanup logic goes here.
    return handlerInput.responseBuilder.getResponse(); // notice we send an empty response
  }
};
/* *
 * The intent reflector is used for interaction model testing and debugging.
 * It will simply repeat the intent the user said. You can create custom handlers for your intents
 * by defining them above, then also adding them to the request handler chain below
 * */
const IntentReflectorHandler = {
  canHandle(handlerInput) {
    return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest';
  },
  handle(handlerInput) {
    const intentName = Alexa.getIntentName(handlerInput.requestEnvelope);
    const speakOutput = `You just triggered ${intentName}. `;

    return handlerInput.responseBuilder
      .speak(speakOutput)
      .reprompt('Plase ask something else.')
      .getResponse();
  }
};
/**
 * Generic error handling to capture any syntax or routing errors. If you receive an error
 * stating the request handler chain is not found, you have not implemented a handler for
 * the intent being invoked or included it in the skill builder below
 * */
const ErrorHandler = {
  canHandle() {
    return true;
  },
  handle(handlerInput, error) {
    const speakOutput = 'Sorry, I had trouble doing what you asked. Please try again.';
```

```
    console.log(`~~~~ Error handled: ${JSON.stringify(error)}`);

    return handlerInput.responseBuilder
      .speak(speakOutput)
      .reprompt(speakOutput)
      .getResponse();
  }
};

/**
 * This handler acts as the entry point for your skill, routing all request and response
 * payloads to the handlers above. Make sure any new handlers or interceptors you've
 * defined are included below. The order matters - they're processed top to bottom
 * */
exports.handler = Alexa.SkillBuilders.custom()
  .addRequestHandlers(
    LaunchRequestHandler,
    FactHandler,
    HelloWorldIntentHandler,
    HelpIntentHandler,
    CancelAndStopIntentHandler,
    FallbackIntentHandler,
    SessionEndedRequestHandler,
    IntentReflectorHandler)
  .addErrorHandlers(
    ErrorHandler)
  .withCustomUserAgent('sample/hello-world/v1.2')
  .lambda();
```