

PedAg-projekti

Testiraportti 1.0

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Koulutusteknologia

Saara Huhta

Viivi Laitinen

Juho Liedes

Maija Turpeinen

Tiivistelmä:

Tämä dokumentti on yksi Jyväskylän yliopiston informaatioteknologian tiedekunnan Koulutusteknologian projekti –kurssilla toteutettavan PedAg-projektin tuotoksista. Testiraportissa kerrotaan projektin testausvaiheeseen valikoituneista alustoista sekä niiden käyttöönotosta käyttötapauksen kouluttamisen kautta. Testiraportti sisältää käyttötapauksen esittelyn lisäksi testausvaiheeseen valikoituneiden alustojen vertailua sekä valinnan ja perustelut pilotointivaiheeseen valitusta alustasta.

Versio	Päivämäärä	Tekijät	Muutokset
0.1	28.3.2023	Huhta, Laitinen, Liedes & Turpeinen	Ensimmäiselle kommenttikierrokselle lähtenyt versio
0.2	5.4.2023	Huhta, Laitinen, Liedes & Turpeinen	Muutokset: tiivistelmä, 1: pieniä lisäyksiä, 1.1: perustelut ensimmäiselle kolmelle testattavalle alustalle, alustakohtaisiin lukuihin (2, 3, 4, 5,) sisällön täydentämistä sekä kuvien yhtenäistämistä ja kaikkiin lisäyksenä yhteenveto alustasta, 6: yhteenveto aloitus
0.3	11.4.2023	Huhta, Laitinen, Liedes & Turpeinen	Muutokset: tiivistelmä: turhan toiston poistaminen, kuvaluettelon lisääminen: kuvaviittausten korjaaminen ja kuvaotsikoiden yhtenäistäminen koko dokumentissa, 2: luvun yhtenäistäminen muihin lukuihin, 2.1.3, 2.1.4 ja 2.2: sisällön täydentäminen, 3.1.1: kielitiedon lisääminen, 4: luvun

			yhtenäistäminen muihin lukuihin sekä integraatio kappaleeseen lähteen lisääminen, 5: luvun yhtenäistäminen muihin lukuihin, 6: täydennystä yhteenvetoon
0.4	12.4.2023	Laitinen	Muutama kirjoitusvirhe sekä muutaman verbin muuttaminen passiiviin, 4.1: hinnoittelutietoon lisäys
0.5	17.4.2023	Laitinen	Lukujen 2 ja 3 ensimmäisten rivien muokkaus, niin että rivitys on järkevä.
1.0	17.4.2023	Laitinen	Dokumentin muuntaminen pdf-muotoon. Sisältöön ei muutoksia.

Taulukko 1. Versiohistoria

Sisällysluettelo

1	Johdanto	7
1.1	Käyttötapaus	7
2	Rasa	9
2.1	Käyttöönotto	9
2.1.1	Kouluttaminen	10
2.1.2	Integraatio	15
2.1.3	Muita huomioita	15
2.2	Yhteenveto Rasasta	16
3	Botpress	17
3.1	Käyttöönotto	17
3.1.1	Kouluttaminen	18
3.1.2	Integraatio	20
3.1.3	Muita huomioita	21
3.2	Yhteenveto Botpressistä	22
4	Juji	23
4.1	Käyttöönotto	24
4.1.1	Kouluttaminen	24
4.1.2	Integraatio	27
4.2	Yhteenveto Jujista	27
5	Kompose Chatbot	29
5.1	Käyttöönotto	29
5.1.1	Kouluttaminen	30
5.1.2	Integraatio	32
5.2	Yhteenveto Kompose Chatbotista	32
6	Yhteenveto	34
	Lähteet	35

Liitteet	37
----------------	----

Kuvaluettelo

Kuva 1 Rasa - Platform (Lähde: Rasa 2023b)	9
Kuva 2 Rasa - Käyttötapauksen domain.yml tiedostosta havainnekuva	11
Kuva 3 Rasa - Käyttötapauksen config.yml tiedostosta havainnekuva	11
Kuva 4 Rasa - Käyttötapauksen rules.yml, stories.yml ja nlu.yml tiedostoista havainnekuva	12
Kuva 5 Rasa - Käyttötapauksen responses havainnekuva	12
Kuva 6 Rasa - Flow-kaavio	14
Kuva 7 Rasa - Arkkitehtuuri (Lähde: Rasa 2023d)	15
Kuva 8 Botpress - Perusnäky	17
Kuva 9 Botpress - Osa käyttötapauksen flow-kaaviosta	18
Kuva 10 Botpress - Agentin vastauksen luonti	19
Kuva 11 Botpress - Esimerkki valmiiksi luodusta koodista	20
Kuva 12 Juji - Platform (Lähde: Juji, 2022a)	23
Kuva 13 Juji - Perusnäky	24
Kuva 14 Juji - Flow-kaavion rakentaminen	25
Kuva 15 Juji - Flow-kaavio	26
Kuva 16 Juji - Agentin viestin luominen	27
Kuva 17 Kompose Chatbot - Perusnäky	29
Kuva 18 Kompose Chatbot - Toimintalogiikka	30
Kuva 19 Kompose Chatbot - Entiteetit	31
Kuva 20 Kompose Chatbot - Flow-kaavio	32

1 Johdanto

Tässä testiraportissa esitellään PedAg-projektin testausvaiheeseen valikoituneita alustoja sekä niiden käyttöönottoa. Kartoituksen perusteella testausvaiheen alustoiksi valikoituivat Rasa, Botpress, Juji, Dialogflow sekä Kompose Chatbot, joista Dialogflow kuitenkin päätettiin jättää lopulta testaamatta. Syy Dialogflown testauksesta pudottamiseen oli, että alustan käyttöönotto oli haastavaa ja olisi vaatinut monipuolista osaamista sekä paljon työtunteja. Testiraportissa kerrotaan neljästä testausvaiheesta testatusta alustasta perustietoja, käyttöönottoon liittyviä asioita, kouluttamisen vaiheita sekä integraatiomahdollisuuksia. Pääpaino testausvaiheessa on ollut niiden ominaisuuksien testaamisessa, jotka arvioitiin merkittäviksi tulevan pilotoinnin näkökulmasta. Testiraportissa kerrotaan alustoista käyttötapauksen kouluttamisen kautta ja avataan sitä, mitkä asiat agentin kouluttamisessa onnistuivat ja missä oli haasteita. Testiraporttiin on kirjattu havainnot testausvaiheesta projektiryhmän jäsenten näkökulmasta sekä täydennetty havaintoja alustojen omista dokumentaatioista.

Testauksen, testiraportin sekä arviointimatriisin avulla on tarkoitus valita yksi alusta projektin toiseen vaiheeseen, jossa pilotoidaan pedagoginen keskusteleva agentti.

1.1 Käyttötapaus

Käyttötapauksen suunnittelu lähti liikkeelle tilaajan esittämistä ajatuksista ja toiveista projektin toisen vaiheen pilotoinnin suhteen. Käyttötapauksella haluttiin saada testattua alustoilta niitä ominaisuuksia, joita tulevassa pilotoinnissa olisi tarkoitus käyttää. Ensimmäinen käyttötapaus haluttiin pitää sisällön puolesta melko yksinkertaisena ja keskittyä ominaisuuksien testaamiseen. Käyttötapauksen flow-kaavio (ks. liite 1) sekä käyttötapauksen tekstidokumentti (ks. liite 2) avaavat yksityiskohtaisemmin sitä, mitä käyttötapauksessa lähdettiin alustoille kouluttamaan. Ominaisuudet, joita alustalla rakennetulta agentilta haluttiin ensimmäisessä käyttötapauksessa testata, on listattu seuraavassa:

- Pystyykö agentti tallentamaan tietoja käyttäjältä? (esimerkiksi nimi)
- Pystyykö agentti syöttämään tekstiä, painikkeita, kuvia, linkkejä, aineistoa?
- Onko käyttäjällä mahdollisuus vastauksen kirjoittamiseen?
- Onko käyttäjällä mahdollisuus avoimen kysymyksen esittämiseen?

Projektiryhmä päätti aloittaa käyttötapauksen kouluttamisen ensin kolmella alustalla: Rasalla, Botpressillä sekä Jujilla. Nämä kolme valikoituivat ensimmäisiksi testattaviksi alustoiksi sen vuoksi, että niihin oli ennen testausvaihetta ennätetty perehtyä hieman enemmän ja niiden käyttöönotto sekä kouluttamisen aloitus

17.4.2023

vaikuttavat selkeämmiltä Kompose Chatbottiin ja Dialogflow:hun verrattuina. Jo kolmen ensimmäiseksi testattavan alustan kohdalla huomattiin, että suunniteltuun käyttötapaukseen oli testausvaiheen aikana tehtävä pieniä alustakohtaisia muutoksia. Ensimmäisen kokonaisen testausviikon aikana havaittiin myös se, että eri alustojen parissa joudutaan käyttämään eri määrä työtunteja käyttötapauksen kouluttamiseksi. Ensimmäisen testausviikon lopuksi lähdettiin kouluttamaan myös neljättä alustaa Kompose Chatbottia, jonka koulutus jatkui myös toisella testausviikolla.

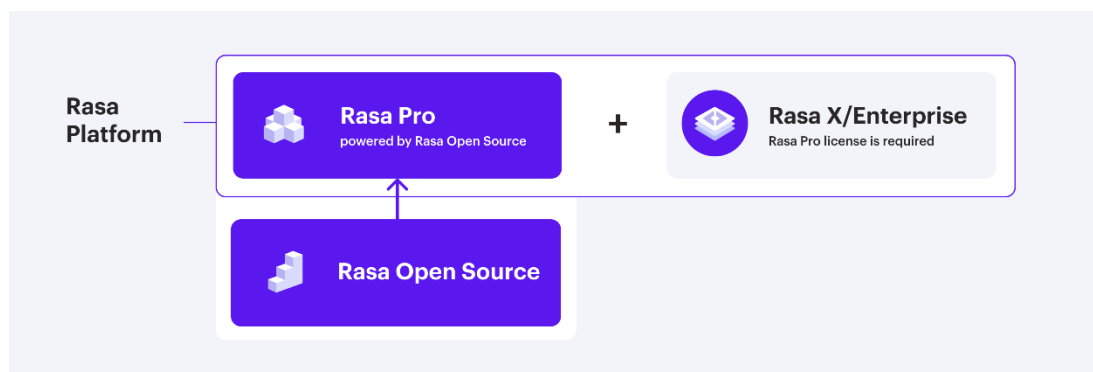
Ensimmäisen testausviikon jälkeen projektikokouksessa keskusteltiin testaamisista ja esiteltiin tilaajille Botpressillä rakennettu agentti. Kokouksen jälkeen tilaajat pääsivät kurkistamaan myös Jujilla sekä Kompose Chatbotilla rakennettuja agentteja videoiden muodossa. Kokouksessa käytyjen keskustelujen sekä kokouksen jälkeen tilaajille esitettyjen kysymysten vastausten pohjalta päädyttiin testausvaiheessa ratkaisuun pudottaa Dialogflow testattavien alustoiden listalta ja jatkaa integraatiomahdollisuuksien selvittelyä parhaiten ensimmäisellä testausviikolla käyttötapauksen luomisessa onnistuneen Botpressin kanssa.

2 Rasa

Tässä luvussa käsitellään alustaa nimeltä Rasa. Testaus on toteutettu Rasa Open Source 3.x -versiolla.

Rasan ensimmäinen versio, Rasa NLU, on julkaistu joulukuussa 2016. Rasa NLU oli avoimen lähdekoodin alusta, joka kehitettiin käyttämään edistyneempää teknologiaa chatbottien rakentamisessa. Rasaa on tämän jälkeen kehitetty ja laajennettu moniin ominaisuuksiin. (Nichol, 2017.) Rasa on ladattu yli 25 miljoonaa kertaa julkaisunsa jälkeen. Rasan päätoimisto on San Franciscossa, Yhdysvalloissa, mutta yhtiö toimii kansainvälisesti myös Berliinissä, Lontoossa, Pariisissa sekä Belgradissa. (Rasa, 2023a.)

Nykyisellään Rasan päätuote on Rasa Platform-alusta (Kuva 1), johon kuuluu Rasa Pro sekä Rasa X/Enterprise. Rasa X/Enterprise versiot vaativat käyttäjältä vain vähäisiä ohjelmointitaitoja. Rasa Pro on vuorovaikutteinen tekoäly framework-alusta, jonka toimintaa rakennetaan Rasa Open Sourcella. (Rasa, 2023a.) Rasan päätuotteet ovat maksullisia. Maksullisten tuotteiden hinnoista ei löydy Rasan nettisivuilta tietoa.



Kuva 1 Rasa - Platform (Lähde: Rasa 2023b)

2.1 Käyttöönotto

Rasalla voi rakentaa ja kehittää teksti- ja äänipohjaisia tekoälyä hyödyntäviä agenteja. Eri tuotteissa on käytettävissä erilaisia teknologioita. Tässä dokumentissa käsitellään ja on perehdytty erityisesti avoimen lähdekoodin Rasa Open Source framework-alustaan, joka on ilmainen. Rasa Open Source frameworkissa ei ole graafista käyttöliittymää vaan ohjelmaa käytetään komentotulkilta. Rasaan tässä testiraportissa viitattaessa tarkoitetaan edellä mainittua versiota, ellei toisin mainita.

Rasasta löytyy kattava dokumentaatio (Rasa, 2023b), jota on projektin tekoaikaan päivitetty viikoittain. Rasan dokumentaatiosta löytyy ohjeet Windowsille, Macille ja Linuxille. Huomioitavaa on, että testaaminen ja integraatioiden selvittely on tehty Windowsin käyttäjän näkökulmasta.

Ennen ohjelman latausta voi kokeilla Rasan toimintaperiaatteita interaktiivisen demoalustan kautta selaimessa. Lataustapoja ja ohjeita on useita. Latauksessa voi hyödyntää Rasan oman dokumentaation ohjetta

(Rasa 2023d). Lataaminen tapahtuu myöhemmin kuvattujen valmisteluiden jälkeen dokumentaatiassa ohjatuilla komennoilla komentotulkissa.

Rasalla on oma ”Rasa Learning Centre” (Rasa, 2023c) eli sivusto, joka tarjoaa käyttäjälle tietoa vuorovaikutteisista tekoäly agenteista ja niiden rakentamisesta Rasalla. Rasalla on myös Youtube-kanava (RasaHQ, 2023b), jonka kautta on saatavilla monia ohjevideoita. Ohje- ja koulutusmateriaalia on saatavilla runsaasti ja lukuisilla eri kielillä myös muilta toimijoilta. Rasa ja sen käyttö on haastava hahmottaa pelkistä tiedostoista, joten jo käyttöönottoaiheessa hyödynnettiin paljon ohjevideoita. Ohjeiden osalta oman haasteensa tuo, että tunnistaa ja osaa hyödyntää ohjeen omassa käytössä olevaan tuotteeseen ja versioon.

Rasan lataaminen ja käyttöönotto vaativat ohjelmia, jotka mahdollistavat Rasan käytön. Ennen Rasan latausta tulee olla käytettävissä Visual Studio (tai vastaava), Python (pip) ja virtuaalinen ympäristö (kuten venv). Käyttöönottoon ja siihen tarvittavaan aikaan vaikuttaa siis myös, mitä ohjelmia ja niiden käyttöosaamista käyttäjällä on ennestään. Huomioitavaa on myös, että erilaiset agentin toiminnot vaativat uusien ohjelmien lataamista tai käyttöönottoa. Esimerkiksi koneoppimisen hyödyntäminen vaatii Pythonin lisäpakettien latauksia, kuten SpaCy tai MITIE. Rasan sekä muiden ohjelmien latauksen yhteydessä haasteena oli ohjelmien ja tiedostojen polutus.

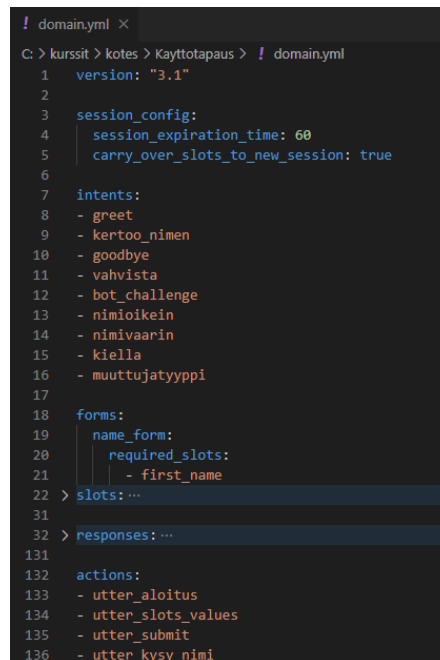
Rasa ohjelmisto sekä sillä rakennettavat agentit sijaitsevat lähtökohtaisesti käyttäjän oman laitteen muistissa. Erilaisten rajapintojen sekä ilmeisesti maksullisen Rasa version kautta on mahdollista hyödyntää myös Rasan ”localhost” tai kolmannen osapuolen palvelinta, jossa tiedostoja voisi säilyttää. Testatessa localhost toimi vain osittain ilmeisesti käytettävän ohjelman version vuoksi. Myöskään ulkoisen pilvipalvelun integrointia ei testattu. Käyttöönotossa huomattiin, että jo alusta alkaen kansiorakenteessa on järkevää ottaa käyttöön jonkinlainen versionhallinta mahdollisuus.

2.1.1 Kouluttaminen

Komentotulkilla agentin koulutus tapahtuu reaaliaikaisesti tai muokkaamalla tiedostoja ja ajamalla ne käyttöön yksi koulutus kerrallaan. Rasan sivuilta pystyy lataamaan tiedostot valmiiseen esimerkkiagenttiin, jonka avulla kouluttamista pääsee harjoittelemaan ja aloittamaan agentin rakentamisen. Useimmat tarvittavat komennot löytyvät selkeästi dokumentaatiosta (Rasa, 2023e). Ohjelmointikieli on pääasiassa Python, mutta koulutuksessa tiedostoihin kirjoitetaan YAML:lla. Agentin testaaminen on vain komentotulkissa, joten painikkeet tai kuvat näkyvät myös vain teksteinä.

Kouluttamisen aloittamiseksi on perehdyttävä huolellisesti dokumentaatioon sekä Rasan koulutusvideoihin (RasaHQ, 2023c). Kouluttaessa tiedostoja muokkaamalla on oleellinen tietää tärkeimmät tiedostot. Pää tiedosto on domain.yml, joka ohjaa kaikkea mitä agentin tulee osata. Domain.yml:ssä (Kuva 2) on

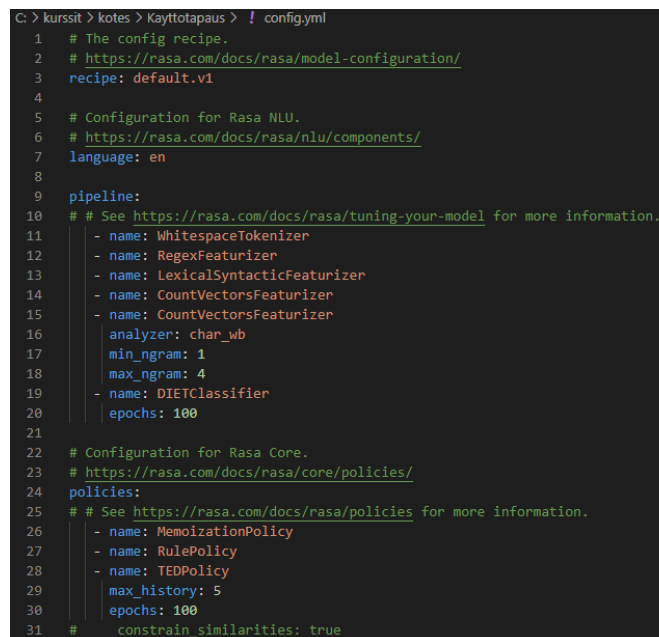
määriteltynä agentin intentit eli aikomukset mitä käyttäjä sanoo, forms eli muodostajat, slotit eli muuttujat, responses eli vastaukset sekä actions eli toiminnot.



```
! domain.yml x
C: > kurssit > kotes > Kayttotapaus > ! domain.yml
1  version: "3.1"
2
3  session_config:
4    session_expiration_time: 60
5    carry_over_slots_to_new_session: true
6
7  intents:
8    - greet
9    - kertoo_nimen
10   - goodbye
11   - vahvista
12   - bot_challenge
13   - nimioikein
14   - nimivaarin
15   - kiella
16   - muuttujatyyppi
17
18  forms:
19    name_form:
20      required_slots:
21        - first_name
22  > slots: ...
31
32  > responses: ...
131
132  actions:
133    - utter_aloitus
134    - utter_slots_values
135    - utter_submit
136    - utter_kysy_nimi
```

Kuva 2 Rasa - Käyttötapausten domain.yml tiedostosta havainnekuva

Päätiedoston lisäksi tärkeitä tiedostoja jo yksinkertaisenkin agentin kouluttamisessa ovat config.yml tiedosto sekä agentin muuta tietoa sisältävät tiedostot nlu.yml, stories.yml sekä rules.yml. Config.yml (Kuva 3) tiedostossa määritellään komponentit ja käytänteet, joita agentti käyttää ennakoidakseen käyttäjän syötteitä.



```
C: > kurssit > kotes > Kayttotapaus > ! config.yml
1  # The config recipe.
2  # https://rasa.com/docs/rasa/model-configuration/
3  recipe: default.v1
4
5  # Configuration for Rasa NLU.
6  # https://rasa.com/docs/rasa/nlu/components/
7  language: en
8
9  pipeline:
10   # See https://rasa.com/docs/rasa/tuning-your-model for more information.
11   - name: WhitespaceTokenizer
12   - name: RegexFeaturizer
13   - name: LexicalSyntacticFeaturizer
14   - name: CountVectorsFeaturizer
15   - name: CountVectorsFeaturizer
16     analyzer: char_wb
17     min_ngram: 1
18     max_ngram: 4
19   - name: DIETClassifier
20     epochs: 100
21
22  # Configuration for Rasa Core.
23  # https://rasa.com/docs/rasa/core/policies/
24  policies:
25   # See https://rasa.com/docs/rasa/policies for more information.
26   - name: MemoizationPolicy
27   - name: RulePolicy
28   - name: TEDPolicy
29     max_history: 5
30     epochs: 100
31   # constrain_similarities: true
```

Kuva 3 Rasa - Käyttötapausten config.yml tiedostosta havainnekuva

17.4.2023

Nlu.yml tiedostoon määritellään aikomukset, mitä käyttäjät sanovat ja näiden koulutusdata. Stories.yml tiedostoon määritellään koulutusdata, joka ohjaa agenttia mitä se tekee seuraavaksi. Rules.yml tiedostoon määritellään lyhyitä osia keskustelusta, jotka menevät aina samoin. Näitä havainnollistetaan kuvassa 4.

```

C: > kurssit > kotes > Käyttötapaus > data > ! rules.yml
1 version: "3.1"
2
3 rules:
4
5 - rule: varmistaa aina että käyttäjä haluaa lopettaa
6   steps:
7     - intent: goodbye
8     - action: utter_lopeta
9     - action: utter_goodbye
10
11 - rule: Say 'I am a bot' anytime the user challenges
12   steps:
13     - intent: bot_challenge
14     - action: utter_iamabot
15
16 - rule: Activate form
17   steps:
18     - intent: request_names
19     - action: name_form
20     - active_loop: name_form
21
22 - rule: Submit form
23   condition:
24     - active_loop: name_form
25   steps:
26     - action: name_form
27     - active_loop: null

C: > kurssit > kotes > Käyttötapaus > data > ! stories.yml
1 version: "3.1"
2
3 stories:
4
5 - story: Perustarina
6   steps:
7     - intent: greet
8     - action: utter_aloitus
9     - action: utter_ask_first_name
10    - action: name_form
11    - active_loop: name_form
12    - slot_was_set:
13      - requested_slot: first_name
14    - slot_was_set:
15      - first_name: vincent
16    - slot_was_set:
17      - requested_slot: null
18    - active_loop: null
19    - action: utter_varmista
20    - action: utter_slots_values
21
22 - story: Nimi meni vaarin
23   steps:
24     - intent: nimivaarin
25     - action: utter_nimiuudelleen

C: > kurssit > kotes > Käyttötapaus > data > ! nlu.yml
1 version: "3.1"
2 nlu:
3 - intent: greet
4   examples: |
5     - moi
6     - hei
7     - hi
8     - hello
9     - terve
10    - moikka
11    - hey
12 - intent: kertoo_nimen
13   examples: |
14     - Minun nimeni on
15     - Nimeni on
16     - olen
17     - oon
18 - intent: goodbye
19   examples: |
20     - lopeta
21     - heippa
22     - moikka
23     - moimoi
24     - moi moi
25     - hei hei
26 - intent: vahvista
27   examples: |

```

Kuva 4 Rasa - Käyttötapausten rules.yml, stories.yml ja nlu.yml tiedostoista havainnekuva

Rasassa voi käyttää entiteettejä, jotka hyödyntävät tekoälyä. Agentin syötteinä (responses, Kuva 5) onnistuvat teksti, painike, kuvalinkki, linkit. Käyttäjän syötteinä onnistuu vastaus painikkeilla tai avoimena tekstinä sekä kysymyksenä.

```

responses:
  utter_aloitus:
    - text: "Hei, olen ViSu-botti ja voin auttaa sinua oppimaan lisää datan visualisoinnista."
      buttons:
        - title: "Lopeta"
          payload: "/goodbye"
  utter_ask_first_name:
    - text: "Mikä on sinun etunimesi?"
      buttons:
        - title: "Lopeta"
          payload: "/goodbye"
  utter_lopeta:
    - text: "Oletko varma, että haluat lopettaa oppimistilanteen?"
      buttons:
        - title: "Lopeta"
          payload: "/goodbye"
        - title: "Palaa"
          payload: "/greet"
  utter_goodbye:
    - text: "Heippa, tervetuloa opiskelemaan uudelleen!"
  utter_varmista:
    - text: "Ymmärsinko nimesi oikein {first_name}?"
      buttons:
        - payload: /nimioikein
          title: Kyllä
        - payload: /nimivaarin
          title: Ei
        - payload: /goodbye
          title: Lopeta

```

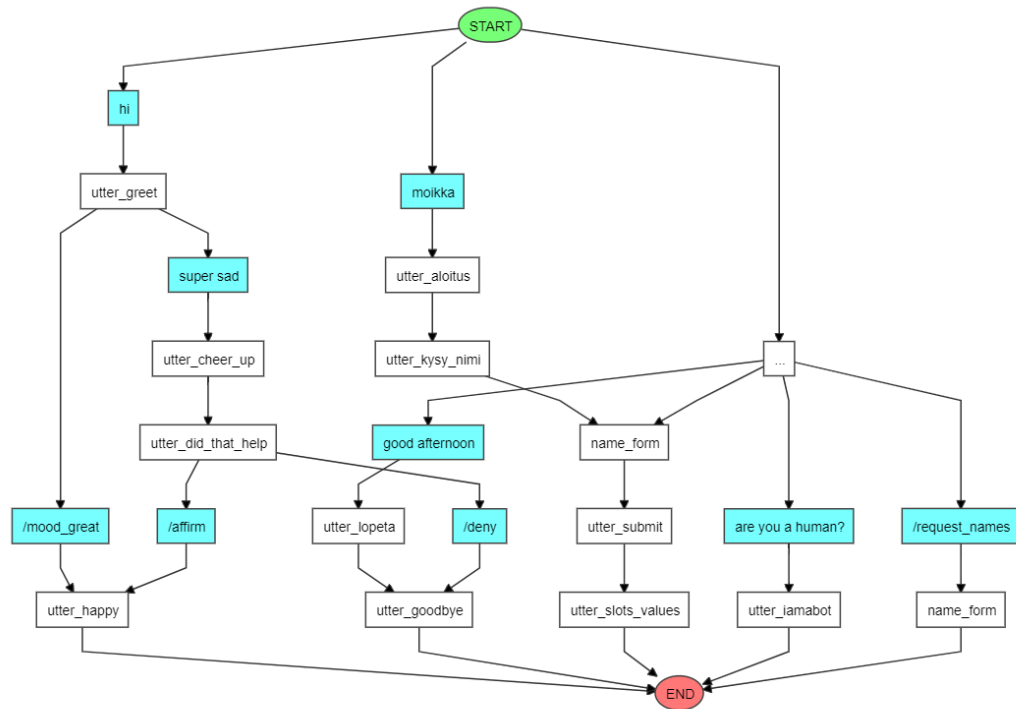
Kuva 5 Rasa - Käyttötapausten responses havainnekuva

Kouluttaminen tiedostoja muokkaamalla on aikaa vievää, sillä jokainen uusi muokkaus pitää tehdä ”rasa train” -komennolla ja tämän ajaminen vei useamman minuutin käytössä olleilla tietokoneilla. Lisäksi kokonaisuuden hahmottaminen oli haastavaa. Jokainen askel tuli suunnitella erittäin huolellisesti, jotta jokaiseen tiedostoon tuli oikeat lisäykset.

Interaktiivisesti kouluttaessa tiedostot päivittyvät automaattisesti, mutta tässä kestää pitkään ja kokonaisuutta on silti haastava hahmottaa. Koulutus on hidasta ja vaatii paljon manuaalista työtä. Interaktiivinen koulutus käynnistetään komentotulkissa, jonka jälkeen käyttäjä ryhtyy keskusteluun agentin kanssa. Agentti generoi omasta mielestään oikean vastauksen perustuen koulutustiedostoihin, jonka jälkeen käyttäjä joko hyväksyy tai hylkää ehdotuksen. Mikäli käyttäjä hylkää ehdotuksen, on sen tarjottava agentille oikea vaihtoehto olemassa olevasta aikomusten tai tapahtumien listauksesta, tai luotava ja määriteltävä uusi. Interaktiivisesti kouluttaessa käyttäjä näkee jatkuvasti muuttujien arvot sekä sen, kuinka suuren oikeellisuuden arvon agentti antaa eri aikomus- tai vastausvaihtoehdoille. Kun koulutus on saatettu päätökseen, käyttäjä poistuu interaktiivisesta tilasta ja alusta tallentaa tuoreen koulutusmateriaalin sekä mahdolliset muut muutokset tiedostoihin.

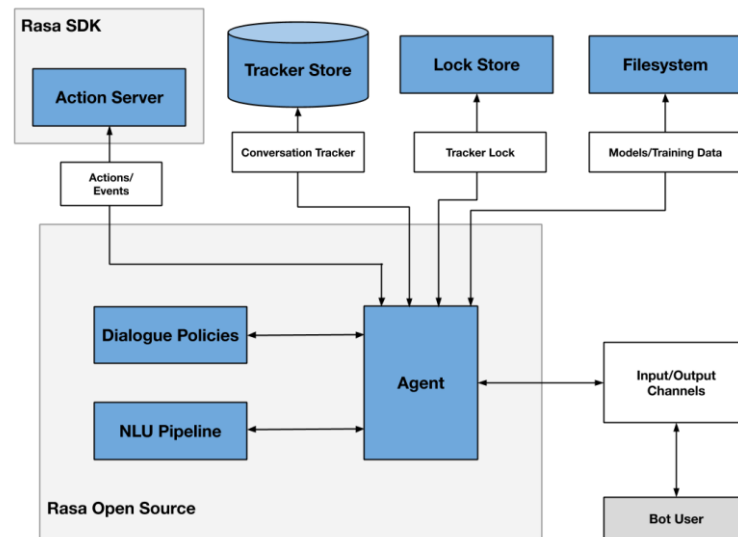
Kouluttamisessa voi hyödyntää myös valmiita ladattavia tiedostoja. Käyttötapauksessa nimen muodostamisessa hyödynnettiin ja soveltaen käytettiin valmista tiedostoa (RasaHQ, 2021). GitHubissa (RasaHG, 2023a) on aktiivinen Rasan kehitysyhteisö, josta on löydettävissä paljon valmiita koodeja. Tammikuussa 2023 Rasan koodiongelmien ratkaisualusta on siirretty Jiraan (Jira, 2023), josta voi myös etsiä ratkaisuja moniin ongelma-kohtiin.

Käyttötapausta testatessa ei saatu toimivaa agenttia aikaiseksi. Käyttötapauksen ylätasoinen suoraviivainen eteneminen saatiin kertaalleen toimimaan, mutta agentti hajosi, kun tähän alettiin rakentamaan nimen tallennusta ja muita toimintoja. Kokonaisuudessaan Rasalla käyttötapauksen rakentaminen oli hyvin aikaa vievää ja haastavaa. Rasasta sai agentin stories-osion visuaalisen rakenteen ajettua näkyviin (Kuva 6) paikalliselle palvelimelle (localhost).



Kuva 6 Rasa - Flow-kaavio

Rasassa voi hyödyntää lukuisia erilaisia teknologioita, mutta se vaatii kehittyneitä ohjelmointitaitoja sekä osaamista rajapinnoista. Rasa tukee esimerkiksi kaikilla kielillä kouluttamista, mutta tämä vaatii, että harjoitusdata on tehtävä manuaalisesti tai on hyödynnettävä esimerkiksi SpaCy:n (kehittynyt NLP) valmista suomen kielen mallia (Rasa, 2023d). Toinen esimerkki hyödynnettävästä kehittyneestä NLP:stä on Tensorflow. Toimintoja voi ja on suositeltavaa rakentaa ja ajaa esimerkiksi kirjoittamalla Pythonilla luokkia erillisiin tiedostoihin. Toimintojen ajaminen erillisissä tiedostoissa tarvitsee Action Serverin ja Action Server puolestaan Rasa SDK:n. Kuva 7 visualisoi Rasan arkkitehtuuria ja kuinka monista erillisistä ohjelmista ja osista agentti tulee rakentumaan.



Kuva 7 Rasa - Arkkitehtuuri (Lähde: Rasa 2023d)

2.1.2 Integraatio

Käyttötapauksen integraatiota verkkosivustoille tai viestiohjelmiin ei ole testattu, sillä mitään toimivaa agenttia ei saatu Rasalla tehtyä. Integraatio vaatii useiden teknologioiden lataamista ja opiskelua. Rasalla rakennetun agentin eli Rasa Assistant integrointia suositellaan hyödyntäen Rasa Helm Chartia (RasaHQ, 2022). Rasa Helm Chart tarvitsee toimiakseen ladattavan Kubernetes, OpenShift tai muun klusterointiin eli aineiston ryhmittelyyn sopivan palvelimen. Tämän lisäksi tulee olla ladattuna ja käyttöönotettuna Rasa Action Server. Verkkosivuille liittäessä hyödynnetään vielä Rasa Chat Widgettiä tai jotain muuta käyttäjän omaa widgettiä eli verkkosivulla näkyvää agenttia ja sen mahdollistavaa ohjelmaa. Rasa Chat Widgetin voi sisällyttää haluamalleen verkkosivulle lisäämällä Rasan dokumentaatiosta löytyvän valmiin script-elementin verkkosivun HTML:ään. Integrointi erilaisille viestintäkanaville onnistuu webhookkeja hyödyntämällä. Webhookkien avulla tiedonvaihtoa voi tapahtua eri ohjelmien eli tässä tapauksessa viestintäkanavan ja Rasan välillä.

2.1.3 Muita huomioita

Testaamisen yhteydessä ei kokeiltu monia Rasan dokumentaatiosta löytyviä mahdollisuuksia. Tässä kappaleessa on nostettu esiin tekoälyä hyödyntäviä teknologioita, joita Rasalla esitetään olevan käytössä.

Erilaisia komponentteja käyttäen voidaan hyödyntää NLU:ta ja NLP:tä myös muun muassa entiteettien laajentamisessa, aikomusten luokittelussa ja vastausten valinnassa. Rasa mahdollistaa myös NLG:n käyttämisen kutsumalla kehittäjän määrittelemää NLG-palvelinta vastausten muodostamisessa (Rasa, 2023g). Rasa tarjoaa myös implementointia erilaisiin "tracker stores":ihin eli keskustelujen tallentamista erilaisiin tietokantoihin, kuten Oraclen, PostgreSQL:n tai SQLiten tietokantoihin (Rasa, 2023h).

Conversation-Driven development (CDD) on parhaita toimintamalleja kehittää keskustelevia tekoäly agentteja. CDD hyödyntää kaikkia käyttäjien käymiä keskusteluja agentin kanssa ja mahdollistaa agentin kehittämisen juuri käyttäjään sopeutuvaan kieleen ja käytökseen. (Rasa 2023f). Rasassa CDD:tä mahdollisestaan Continuous Integration (CI) ja Continuous Deployment (DC) pipeline eli kanavan kautta. Dokumentaatiosta löytyy ohjeet tähän GitHub actionsia hyödyntäen.

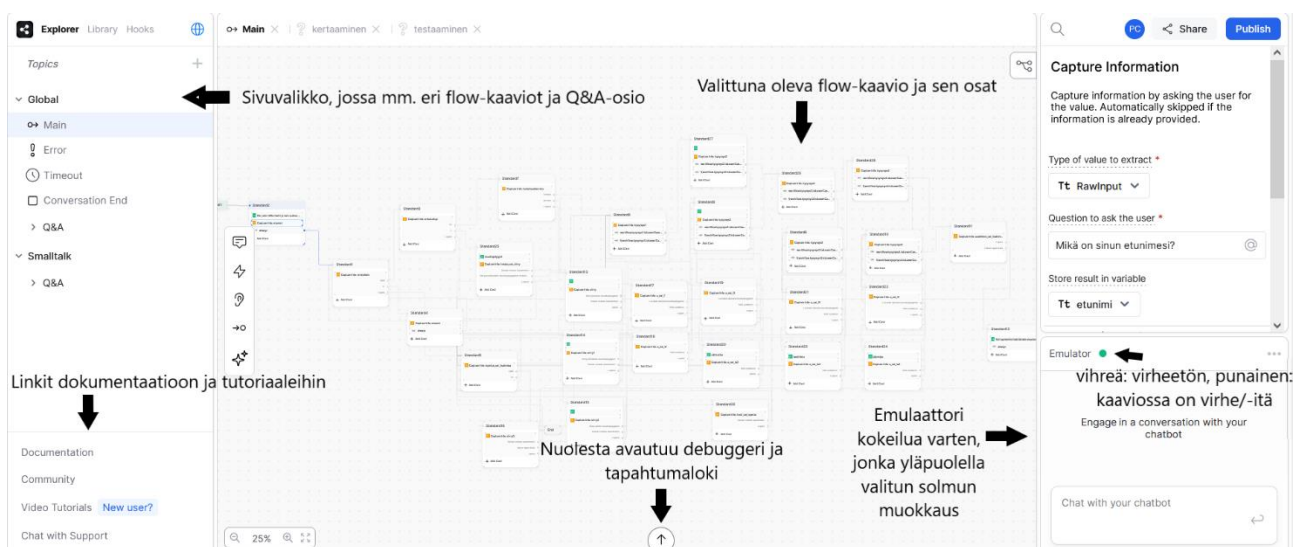
2.2 Yhteenveto Rasasta

Käyttötapausten testaaminen Rasalla ei onnistunut, sillä alusta oli projektiryhmälle vaikeakäyttöinen ja toteuttaminen oli suhteessa olevaan aikaresurssiin liian hidasta. Rasan käyttöönotto vaati jo paljon aikaresurssia ja perehtymistä suhteessa muihin testattuihin alustoihin. Rasa alustan käyttöön tarvitaan hyvää ohjelmointiosaamista, ymmärrystä rajapinnoista ja integroitavista ohjelmista sekä paljon aikaresurssia. Yleisesti kuitenkin dokumentaation ja erilaisilta sivustoilta havaitun kommentoinnin perusteella Rasalla on mahdollista rakentaa monipuolisesti erilaisia tekoälyn teknologioita hyödyntäviä agentteja. Huomioitavaa myös on, että Rasa Open Source framework-alusta on ainut testausvaiheessa käytetty täysin ilmainen alusta.

3 Botpress

Tässä luvussa käsitellään alustaa nimeltä Botpress. Testaus on toteutettu Botpress Cloud –selainversiossa. Koneelle ladattava versio voi poiketa toiminnoiltaan selainversioon verrattuna.

Botpress on perustettu Kanadassa vuonna 2017. Sillä on ollut yli 300000 käyttäjää vuoteen 2023 mennessä, ja alusta on jatkuvan kehityksen ja uudistuksen alla (Businesswire, 2023). Alusta tarjoaa paljon ominaisuuksia ja mahdollisuuksia kouluttaa agentteja ilman koodin kirjoittamista, mutta osa elementeistä vaatii JavaScriptin käyttämistä. Alustasta on saatavilla kaksi versiota: Self-serve ja Enterprise. Self-serve on maksuton, mutta siinä on rajoituksia, kuten ilmaisten viestien ja kehityksessä olevien agenttien rajoitettu määrä. Lisäksi Enterprise-versiossa saa henkilökohtaisempaa ja 24/7-saatavilla olevaa tukea sekä muita yrityksille tärkeitä etuja, kuten SSO-integraatioita ja palvelutasosopimuksia. (Botpress, 2023e.) Botpress API on vielä kehitysvaiheessa, mutta sitä voidaan hyödyntää selaimen Botpress Cloudissa agenttien luomisessa, julkaisemisessa ja ajamisessa (Botpress, 2023f). Botpress nimeää dokumentaatioonsa ominaisuuksikseen muun muassa SDK:n, turvallisuuden, monikanavaisen ja monikielisen tukemisen, vuorovaikutteisen tekoälyn sekä jatkuvan testauksen, kehityksen ja julkaisun mahdollisuudet (Botpress, 2023a). Kuvassa 8 näkyy Botpress Cloud –selainversion perusnäkö, kun on siirtynyt kehittämään valittua agenttia.



Kuva 8 [Botpress - Perusnäkö](#)

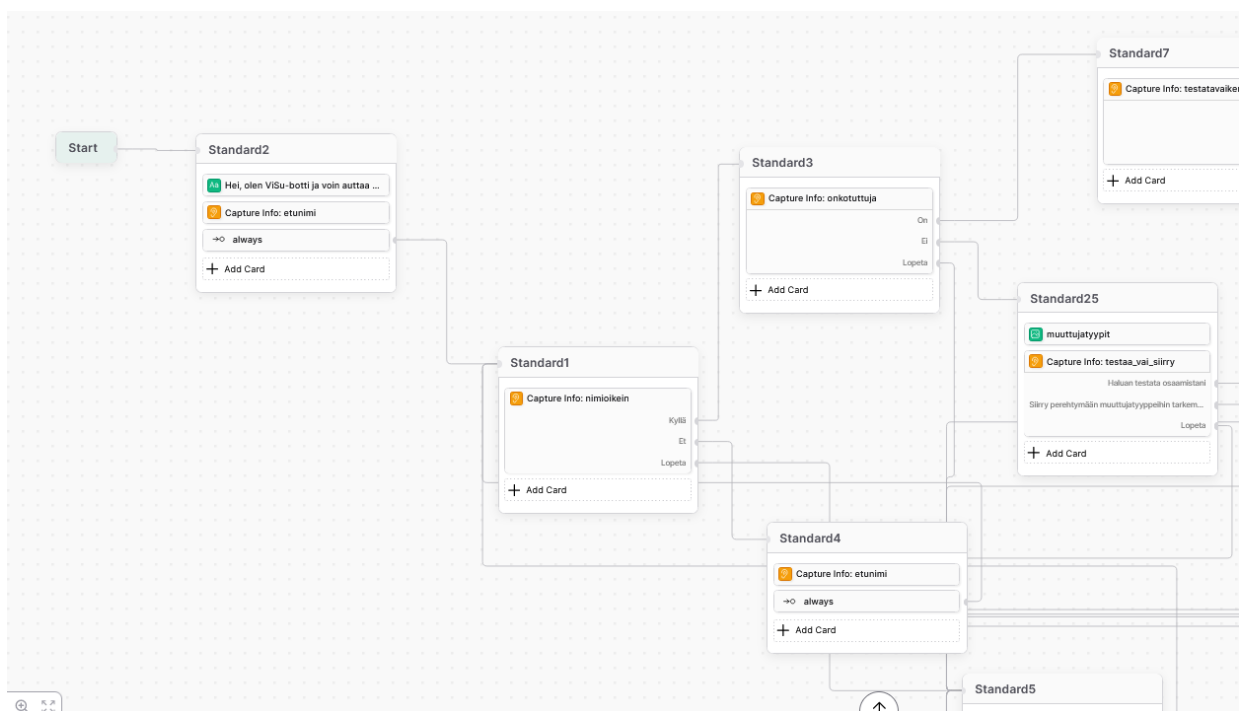
3.1 Käyttöönotto

Botpress on avoimen lähdekoodin platform-alusta, jossa on graafinen käyttöliittymä. Alustaa voi käyttää selaimessa, tai sen voi ladata omalle koneelle. Selainversio on vielä kehitysvaiheessa, mutta on jo toimiva kokonaisuus, jolla voi luoda ja julkaista toimivan agentin. Alustan dokumentaatio on vielä kesken, mutta sitä kehitetään jatkuvasti. Botpress on ilmainen tiettyyn viestimäärään saakka kuukausittain (1000 saapuvaa

viestiä), jonka jälkeen lisäviestit maksavat \$0.005/kpl. Kirjautuessa palveluun ei tarvitse syöttää korttitietoja. Botpressin käyttöön ja agentin kouluttamiseen saa tukea asiakaspalvelusta, joka toimii sähköpostitse sekä livechatissa, ja on avoin kaikille kehittäjille. Lisäksi Botpress järjestää käyttäjilleen erilaisia webinaareja ja tarpeen mukaan myös koulutustilaisuuksia, joita voi pyytää alustan kehittäjiltä. Ilman ylimääräisiä maksuja voi kehittää viittä agenttia kerralla ja mikäli haluaa useamman kuin viisi, on siirryttävä maksulliseen Enterprise-versioon.

3.1.1 Kouluttaminen

Agenttia voidaan lähteä rakentamaan ja kouluttamaan tyhjällä pohjalla, tai tueksi voidaan ladata valmis malli, jota voi muokata omien tarpeiden mukaisesti. Koulutus tapahtuu reaaliaikaisesti, kun käyttäjä tekee muutoksia agentin flow-kaavioon (kuva 9). Alusta tukee erilaisia syötemuotoja, kuten avointa tekstiä ja painikkeita, jotka ovat valittavissa uutta solmua (*node*) luodessa. Solmujen välille luodaan yhteyksiä ja yksi solmu voi viedä useampaan eri solmuun käyttäjän syötteen mukaan. Solmuun voidaan myös luoda ehtoja, jolloin siirrytään tiettyyn solmuun ehdon täytyessä. Solmuissa etenemistä voidaan hallita aikomusten avulla, jolloin tiettyyn solmuun siirrytään nimetyn aikomuksen esiintyessä. Mikäli keskustelussa tulee virhetilanne, ohjautuu se suoraan erilliseen virhettä käsittelevään flow-kaavioon. Myös ”pääkeskustelussa” voi olla useampia flow-kaavioita, jolloin etenemiseen ja sen suunnitteluun saadaan selkeyttä. Virhetilanteen tapaan lopetus voidaan käsitellä omassa kaaviossaan.



Kuva 9 Botpress - Osa käyttötapauksen flow-kaaviosta

17.4.2023

Agentin voi kouluttaa syöttämään käyttäjälle vastauksen kuvana, tekstinä, äänenä, tiedostona, pudotusvalikkona, karusellivalikkona, kuvan ja tekstin yhdistävänä korttina, tai sijaintitietona. Kuvan koko on melko pieni, sillä se mukautuu agentin vastausikkunan kokoon. Käyttäjä puolestaan pystyy mitä ilmeisemmin antamaan vain tekstisyötteen tai vastauksen painamalla vastauspainiketta. Agentin voi myös kouluttaa keräämään tietoa käyttäjän syötteestä, jolloin tiedon voi halutessaan tallentaa valittuun ja määriteltyyn muuttujaan. Etenemiseen liittyviä ehtoja voidaan luoda esimerkiksi näiden muuttujien ja niiden arvojen pohjalta. Tieto voidaan kerätä muun muassa valikoista, boolean-arvoihin perustuvista kytkimistä tai vapaasta tekstisyötteestä, ja vapaasta tekstisyötteestä kerättävän tiedon tyyppi (esim. puhelinnumero, mitta, osoite, lämpötila, väri) voidaan määritellä etukäteen.

The screenshot shows the configuration for a 'MultiChoice' action in Botpress. The interface includes the following sections:

- Type of value to extract ***: A dropdown menu set to 'MultiChoice'.
- Question to ask the user ***: A text input field containing 'Valitettavasti tämä meni väärin.'
- Store result in variable**: A dropdown menu set to 'Select a variable'.
- Choices (3 items)**: A section with a table of choices.

Items	{x} +
Lue lisää oikeasta muuttujatyypistä	→ -
Yritä uudelleen	→ -
Lopeta	→ -
- Extract from History**: A section with a text input field for 'Number of messages from history to extract from' set to '0'.
- Validation**: A section with a dropdown menu.
- Retries**: A section with a dropdown menu.
- Cancellation**: A section with a dropdown menu.

Kuva 10 Botpress - Agentin vastauksen luonti

Agenttia voidaan myös kouluttaa kirjoittamalla solmuihin JavaScriptia, jonka avulla voidaan esimerkiksi määritellä muuttujien arvoja, hallinnoida tapahtumia, tai tehdä validointeja. Ulkoisille API-rajapinnoille voidaan tehdä kutsuja Botpressin tukemilla axios-metodeilla. Koodin kirjoittamisen tukena on alustalla toimiva tekoäly, joka kirjoittaa koodin käyttäjän puolesta annetun ohjeen ja muuttujien mukaan, käyttäjän niin halutessaan (kuva 11). Koodia voi muokata virheiden ilmetessä, mutta alusta luo tekoälyllä valmiin ja usein myös toimivan pohjan.

Execute Code

Change variable values, view event details, custom validations, etc.

Label

teach about data visualization

Code *

```
01 // Define the search query
02 const query = 'data visualization'
03
04 // Define the parameters for the search API
05 const params = {
06   q: query,
07   format: 'json',
08   origin: '*'
09 }
10
11 // Define the URL for the search API
12 const url = 'https://en.wikipedia.org/w/api.php'
13
14 // Make the API request using axios
15 const response = await axios.get(url, { params })
16
17 // Extract the page ID of the first search result
18 const pageId = _.get(response, 'data.query.search[0].pageid')
19
20 // If a page ID was found, get the page content
```

Kuva 11 Botpress - Esimerkki valmiiksi luodusta koodista

Käyttötapauksen vaatimat tekniset ominaisuudet eli käyttäjän avoimen vastauksen lukeminen, painikepohjaisen vastauksen tarjoaminen, agentin kuvamuotoinen vastaaminen sekä nimimuuttujan tallentaminen voidaan toteuttaa Botpressillä. Lisäksi alustalla voidaan käsitellä tiedostoja (agentti voi syöttää sellaisen käyttäjälle) ja Botpressillä on monipuoliset tekoälymahdollisuudet koulutukseen ja kehittämiseen. Agenttia voidaan kokeilla alustan omalla emulaattorilla koulutuksen lomassa, ja testauksen aikana voi seurata tapahtumia, mahdollisia virheilmoituksia ja muuttujien arvoja samassa näkymässä olevista lokeista. Botpressillä voi rakentaa agentin millä tahansa kielellä, mutta NLU-ominaisuuksia vapaamuotoisessa keskustelussa voi hyödyntää tällä hetkellä ainoastaan englanniksi. Suomi on monen muun kielen kanssa listalla tulevista NLU:n tukemista kielistä, joita kehitetään parasta aikaa. (Botpress, 2023g.)

3.1.2 Integraatio

Botpressillä tehty agentti voidaan integroida useaan eri kohteeseen. Agentin voi integroida osaksi viestintäkanavaa, kuten WhatsAppia tai Teamsiä, tai sen voi viedä vapaavalintaiselle verkkosivulle. Jotta agentti voidaan integroida, on se ensiksi julkaistava Botpress Cloud Dashboard –palveluun. Verkkosivulle tehdyssä integroinnissa alusta konfiguroi valmiin HTML-skriptin, jonka avulla agentti voidaan upottaa osaksi kohdesivustoa. Valmiiksi konfiguroitu skripti on valmis HTML:n script-elementti, joka liitetään osaksi kohteen

HTML-tekstiä tai vaihtoehtoisesti omana HTML-elementtinään. Tätä skriptiä ei voi muokata itse, vaan kaikki tieto tulee Botpressin omista tiedostoista. Sen sijaan alustan toista tarjoamaa skriptiä voi muokata itse. Botpress ei suosittele tämän version käyttöä, sillä skriptiä on aina käytävä päivittämässä, kun tekee muutoksia agenttiin, kun taas valmiiksi konfiguroitu skripti tekee tämän käyttäjän puolesta. Tämä versio kuitenkin antaa mahdollisuuden muokata HTML-pätkää ja esimerkiksi lisätä tyylitiedostoja tai omia Reactilla luotuja GUI-viestejä (Botpress, 2023a).

Botpressillä on integraation dokumentaatio vielä kesken, joten kaikkiin integroinnin kohteisiin ei ole vielä tarkkoja ohjeita saatavilla. Integrointiin tarvittava teknologia riippuu kohteesta, ja esimerkiksi Teams-palveluun tehtävä integrointi vaatii Microsoft Azuren käyttöä. Jokaisella viestintäkanavalla on omat konfigurointitapansa, ja useimmat kohteet vaativat erilaisten id-numeroiden ja tunnisteiden käyttöä. Lisäksi on tyypillistä vaatia webhook-konfigurointi viestinvälitystä varten, johon dokumentaatio myös tarjoaa kohdekohtaiset ohjeet (Botpress, 2023b). Julkaisun ja integroinnin jälkeen kehittäjä voi seurata agentin viestijä käyttäjämääriä alustan analytiikkaosion kautta. Botpress Web Chat, eli verkkosivuintegrointi, hostataan AWS S3:lla tai BPFS:llä (Botpress, 2023a). Botpressillä tehty agentti lähtökohtaisesti tukee kaikkia aiemmin mainittuja agentin syötemuotoja, kuten kuvia, videoita ja tiedostoja, mutta näissä on joitain kanavakohtaisia eroja (Botpress, 2023b).

3.1.3 Muita huomioita

Botpressiä testatessa huomattiin, että alusta on päivittänyt GPT-teknologiaan (*Generative pre-trained transformer*) perustuvia ominaisuuksiaan ja lanseerannut uudistetun version palvelustaan. Botpress Community (2023) on tiedottanut, että päivitetty GPT-pohjaiset tekoälyratkaisut hyödyntävät uusimpia LLM-malleja ja tuovat agentteihin laadukkaampaa vuorovaikutteisuutta, sekä tehostavat agentin kehitystä merkittävästi. Päivityksen myötä ChatGPT:tä voidaan heidän mukaansa hyödyntää ratkaisujen etsimisessä sekä kehittäjille että agentin käyttäjälle. Uudistuksen lomassa Botpress myös julkaisi Botpress Go –ratkaisun, joka kykenee lyhyessä ajassa luomaan FAQ-tyyppisen agentin annetun verkkosivun sisällöstä. Lisäksi Botpress on ilmoittanut julkaisevansa lähitulevaisuudessa ominaisuuden, jonka avulla keskustelua voidaan ohjata käyttäjän syötteen lisäksi myös tunnetilan mukaan (Botpress, 2023c). Botpressillä on alkuvuodesta 2023 päivitetty datankeruu- ja tietosuojasuunnitelma, joka löytyy alustan omilta nettisivuilta (Botpress, 2023d).

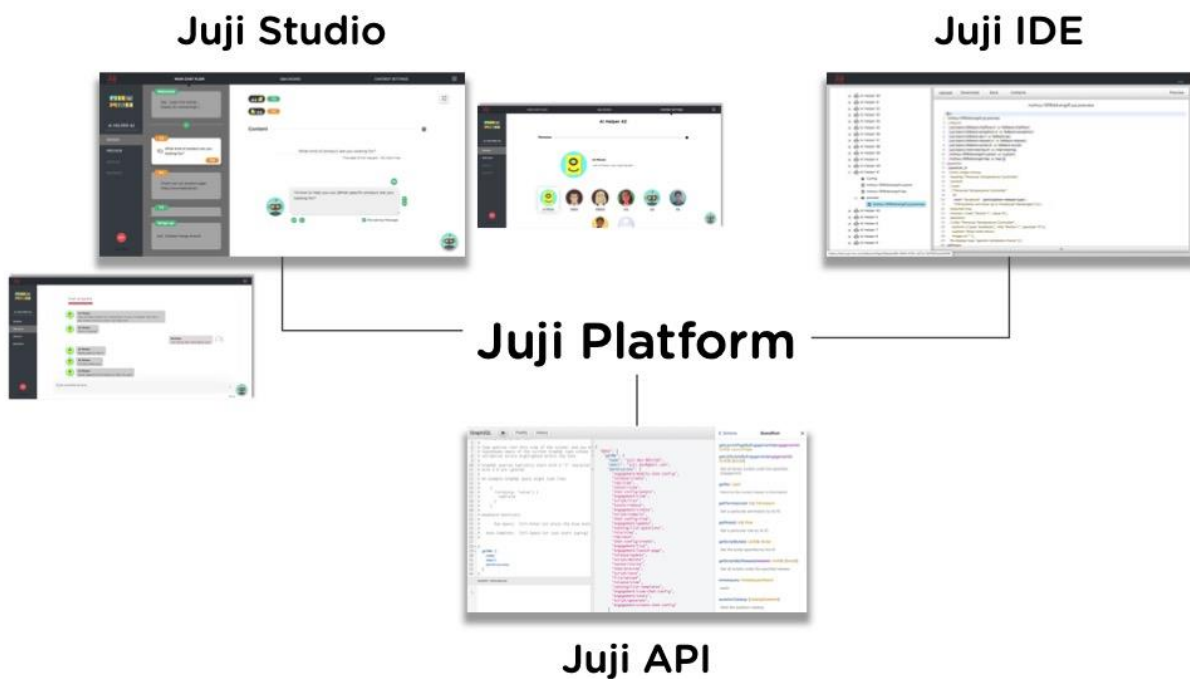
Käyttötapauksen testausvaiheessa uudistuksen mukana tulleita ominaisuuksia ei ole hyödynnetty, sillä päivitys on vasta julkaistu. Tähän liittyvät huomiot näin ollen perustuvat pitkälti Botpressin omiin julkaisuihin. Botpressiä arvioidessa on huomioitava, kuten muidenkin alustojen kohdalla, että kaikkia ominaisuuksia ei olla käyttötapauksen yhteydessä testattu, joten täyttä kokeiluun perustuvaa varmuutta ei ole kaikista ominaisuuksista.

3.2 Yhteenveto Botpressistä

Botpress on kokonaisuudessaan käyttäjäystävällinen, sillä se on vaivaton ottaa käyttöön ja tukea saa dokumentaatiosta, ohjevideoista sekä asiakaspalvelusta. Lisäksi alustan kustannusmalli on kohtuullinen, sillä agentteja voi kehittää ja julkaista ilmaiseksi yhdessä kahdesta eri versiosta. Alusta tarjoaa monipuolisia mahdollisuuksia agentin kouluttamiseen, ja toiminnot hyödyntävät yhä enenevässä määrin erilaisia tekoälyominaisuuksia. Botpress on jatkuvan kehityksen alla, ja dokumentaatiota hiotaan tarpeen mukaan. Alustan testausvaiheessa kaikkia uusimpia tekoälyominaisuuksia ei ollut vielä saatavilla, joten käyttötapaus testattiin ja saatiin toimivaksi sen hetkisillä ominaisuuksilla. Integraatiota testattiin verkkosivulle ja Moodleen, joihin agentti saatiin onnistuneesti upotettua. Tarkempia kokeiluja esimerkiksi tyylitiedoston kanssa ei tehty, kuten ei myöskään integraatioita viestintäkanaviin.

4 Juji

Tässä luvussa käsitellään alustaa nimeltä Juji. Juji on USA:ssa vuonna 2014 perustettu yritys, joka tarjoaa mahdollisuuden rakentaa vuorovaikuttavia agentteja tekoälyä hyödyntäen. Jujin käyttäjämäärästä ei ole saatavilla tietoa. Juji (kuva 12) koostuu kolmesta eri osiosta: Juji studiosta, Juji IDE:stä sekä Juji API:sta. Studio mahdollistaa agenttien rakentamisen, IDE on kehitysympäristö agenttien testaamiseen ja API mahdollistaa Jujilla rakennettujen agenttien integroimisen kolmannen osapuolen järjestelmiin.

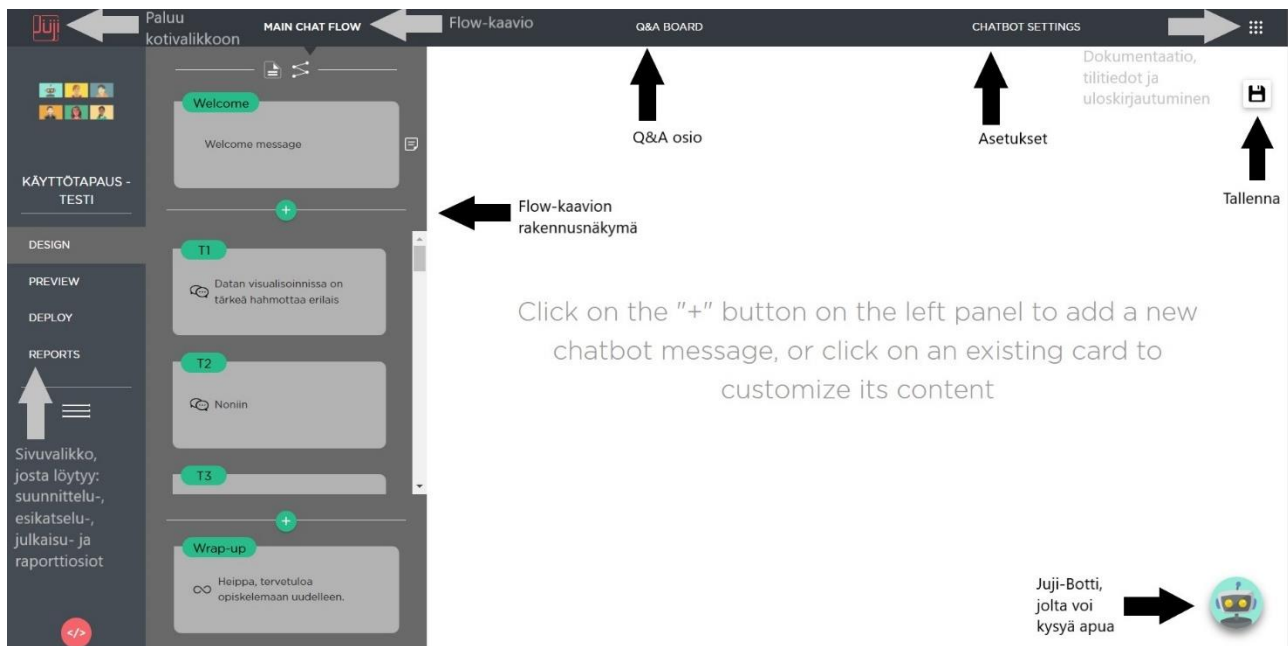


Kuva 12 Juji - Platform (Lähde: Juji, 2022a)

Jujin avulla voi nopeasti ja helposti luoda, käynnistää ja hallita tehokkaita, mukautettuja Juji AI –avustajia chatbottien muodossa, joissa on sisäänrakennettu kognitiivinen älykkyys eli ne pystyvät hyödyntämään tekoälyä (Juji, 2022a).

Jujilta löytyy myös kattava tietosuojasuunnitelma. Juji ei kerää aktiivisesti yhteystietoja käyttäjiltä, mutta chatbottien luojat voivat itse päättää haluavatko kysyä niitä. Chat lokitiedot, joita Juji säilyttää on suojattu alan standardien mukaisilla protokollilla. (Juji, 2022b)

Alustan testaus ja agentin rakentaminen toteutettiin Juji studiossa. Kuvassa 13 on kuva Juji studion perusnäkymästä, joka avautuu, kun siirtyy studioon.



Kuva 13 [Juji - Perusnäköymä](#)

4.1 Käyttöönotto

Juji studio toimii selaimessa ja on graafisen käyttöliittymän omaava Platform-alusta eli käyttäminen ei vaadi koodaustaitoja. Käyttöönotto on nopeaa sillä alustan koneelle lataamista ei tarvita. Juji studiossa voi rakentaa agentteja ilmaiseksi niin monta kuin haluaa. Maksu vaaditaan vasta siinä kohtaa, kun agentin haluaa julkaista. Julkaisun jälkeen Jujin käyttö maksaa 100\$/kk, mutta Jujin dokumenteista ei selviä se, onko maksu yhden agentin hinta vai voiko sillä tehdä useampia agentteja.

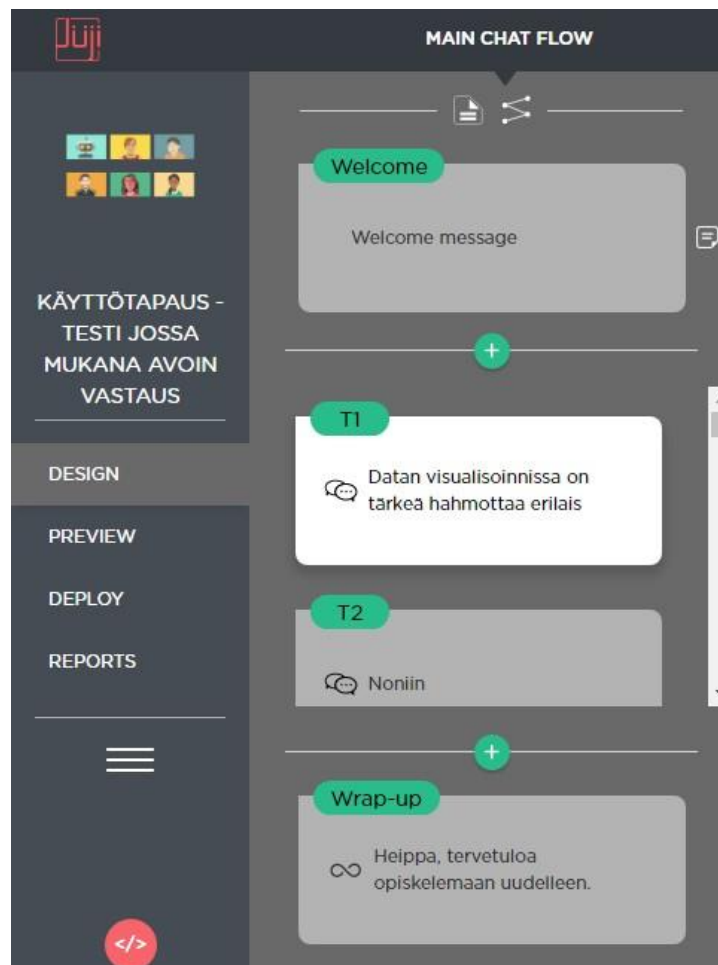
Juji tarjoaa kattavan tekstimuotoisen ohjepaketin ja dokumentaation siitä, kuinka päästä alkuun oman agentin luomisessa. Myös ohjevideoita eri kouluttamisen vaiheista löytyy paljon. Juji studiossa Juji-botti opastaa käyttäjää hänen niin halutessaan vaihe vaiheelta oman agentin luomisessa.

4.1.1 Kouluttaminen

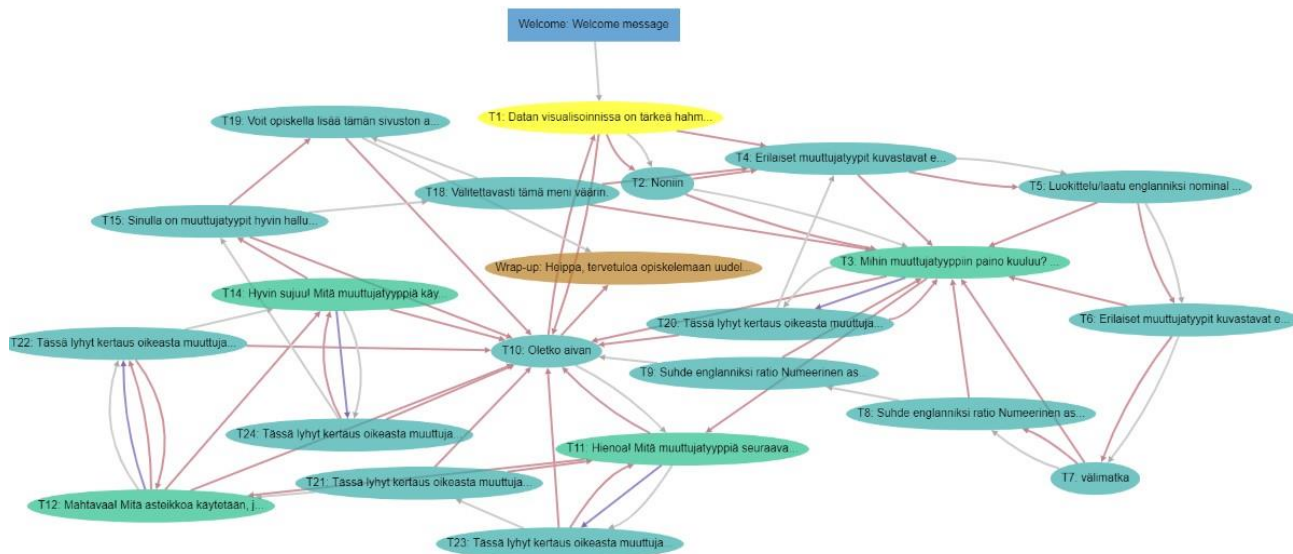
Juji studiossa agentin kouluttaminen on mahdollista aloittaa lähes täysin tyhjälle pohjalle tai valita jokin valmiista pohjista. Vaikka aloittaa luomisen tyhjälle pohjalle, on pohjassa muutama valmis elementti ja esimerkiksi käyttäjän nimi tallentuu automaationa agenttiin. Tämä toi ensimmäisen haasteen käyttötapausten kouluttamisessa. Käyttötapausten ensimmäiset kysymykset nimen tiedustelusta jouduttiin jättämään pois, sillä nimi oli jo tallentuneena agenttiin ensimmäisen testauksen jälkeen, eikä sen nollaaminen onnistunut.

Agentin kouluttaminen tapahtuu reaaliaikaisesti eräänlaista flow-kaaviota (kuva 14) rakentamalla ja muokkaamalla. Agenttia voi kouluttaa millä tahansa luonnollisella kielellä, mutta pohjakielenä toimii englanti

ja alustan käyttö tukee parhaiten sitä. Alusta tukee erilaisia syötemuotoja, kuten tekstiä ja painikkeita, jotka on valittavissa uutta pyyntöä (request) luodessa. Pyyntöjen välille luodaan yhteyksiä ja yksi pyyntö voi viedä useampaan eri pyyntöön käyttäjän syötteen mukaan. Pyyntöön voidaan myös luoda ehtoja, jolloin siirrytään tiettyyn pyyntöön ehdon täytyessä. Kouluttaminen lähtee liikkeelle tervetulopyynnön muokkaamisesta ja jatkuu sen jälkeen muiden pyyntöjen muotoiluun. Näistä rakentuu kouluttamisen edetessä flow-kaavio (kuva 15).

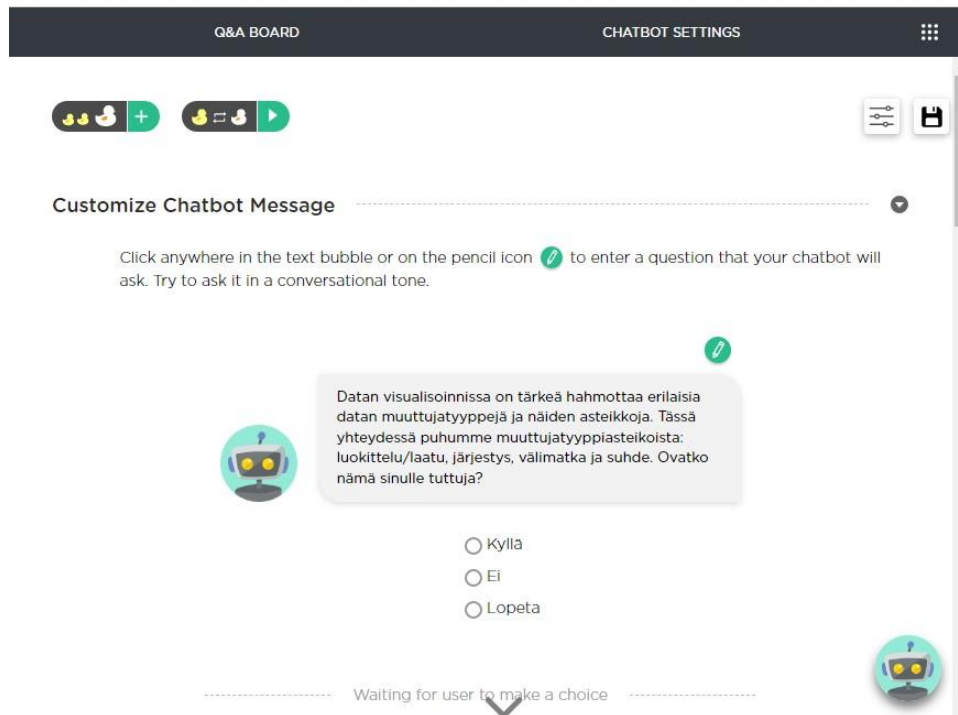


Kuva 14 Juji - Flow-kaavion rakentaminen



Kuva 15 Juji - Flow-kaavio

Agentin viestien (kuva 16) ja kysymysten kouluttaminen onnistuu helposti. Erityisesti painikepohjaisena, jolloin agentti koulutetaan tarjoamaan painikkeita vastausvaihtoehtoina, kouluttaminen on helppoa. Jujilla agentin pystyy kouluttamaan viestimään myös tekstinä, kuvina ja linkkeinä, jotka olivat käyttötapauksen testaamisessa toivottuina ominaisuuksina. Tekstivastauksen luominen on hieman monimutkaisempaa kuin painikkeiden. Kuvien ja linkkien lisääminen koulutuksessa onnistuu kätevästi, sillä graafinen käyttöliittymä itsessään ohjaa ja neuvoo siihen. Käyttötapauksella koulutetulta agentilta toivottiin, että se pystyisi syöttämään käyttäjälle tiedoston. Tätä ei kuitenkaan saatu onnistuneesti toteutettua. Käyttäjän syötteen on mahdollista olla Juji studiossa tekstiä tai painikkeen painallus. Esimerkiksi äänen tallennus ei ole mahdollista.



Kuva 16 Juji - Agentin viestin luominen

Kouluttamiseen liittyvä Q&A osio eli kysymykset, joita käyttäjältä oletetaan ja vastaukset, joita agentin halutaan antavan, on mahdollista kirjoittaa studiossa yksitellen tai ajamalla ne erillisenä tiedostona yhdellä kertaa alustalle. Toimialuekohtaisia Q&A malleja löytyy Jujilta myös valmiina, joten kaikkea ei tarvitse kirjoittaa itse.

4.1.2 Integraatio

Testausvaiheessa ei päästy testaamaan Jujin integrointimahdollisuuksia, sillä agentin julkaisemisen jälkeen alusta muuttuu maksulliseksi ja integroinnin testaus olisi edellyttänyt agentin julkaisemista. Jujilla rakennettu agentti on kuitenkin mahdollista integroida verkkosivulle tai Facebook Messengeriin suoraan Juji studiosta. Juji API:n avulla agentin integroiminen on mahdollista myös kolmannen osapuolen sovelluksiin.

Juji API perustuu GraphQL:ään, jonka avulla sovellus voi kysellä tietoja Juji-chatbot-palvelimelta. Juji API:a on tarvetta käyttää, jos yritys haluaa brändätä chatbotin keskusteluikkunan tai viestit ja chatbotin on oltava osa mobiili- tai verkkosovellusta. (Juji, 2022c)

4.2 Yhteenveto Jujista

Juji on monipuolisuutensa vuoksi varteenotettava alustavaihtoehto, jos haluaa rakentaa pedagogisen agentin. Se on helppo ottaa käyttöön ja sillä agentin rakentaminen on pääosin selkeää. Huonona puolena Jujissa on sen maksullisuus ja se, että sen julkaisemista ja integraatiomahdollisuuksia ei pääse kunnolla testaamaan

17.4.2023

ennen kuukausimaksun suorittamista. Juji studion, Juji IDEn sekä Juji API:n yhteistyöllä agentin rakentamisen, testauksen ja julkaisun pitäisi kuitenkin pystyä onnistuneesti suorittamaan ilman muita integroituvia ohjelmia tai koodinkirjoittamista. Tämä lisää todennäköisesti käyttäjämäärää ja mahdollisesti kasvattaa Jujin suosiota tulevaisuudessa, mikäli yritys pystyy muutoin kehittämään ja pitämään toimintansa kilpailukykyisenä muihin vastaaviin yrityksiin verrattuna.

5 Kompose Chatbot

Tässä luvussa käsitellään alustaa nimeltä Kompose Chatbot. Kompose Chatbotin on luonut Kommunicate-niminen yritys, joka sijaitsee Intiassa. Sillä on toimipiste myös Yhdysvalloissa. Kommunicate on perustettu vuonna 2020. Yritys julkaisi pian perustamisensa jälkeen Kompose Chatbotin.

Kompose Chatbotista on saatavilla sekä selainversio että omalle koneelle asennettava versio. Sen kotisivuilla (Kommunicate, 2023b) mainittuja siihen sisältyviä teknologioita ja ominaisuuksia ovat muiden muassa itseoppiva tekoäly (NLP mainittu), valikoima valmiita pohjia agenteille, mahdollisuus ohjata käyttäjä keskustelemaan ihmisen kanssa, rajaton määrä agenteja flow-kaavioineen, GDPR-hyväksytty tietosuoja ja ympärivuorokautinen tuki. Kompose Chatbot tukee kaikkia luonnollisia kieliä, mutta englanti on parhaiten tuettu. Käytettävissä olevat ominaisuudet kuitenkin vaihtelevat eri tilausvaihtoehtojen välillä (Kompose Chatbot, 2023b).



Kuva 17 [Kompose Chatbot - Perusnäköymä](#)

5.1 Käyttöönotto

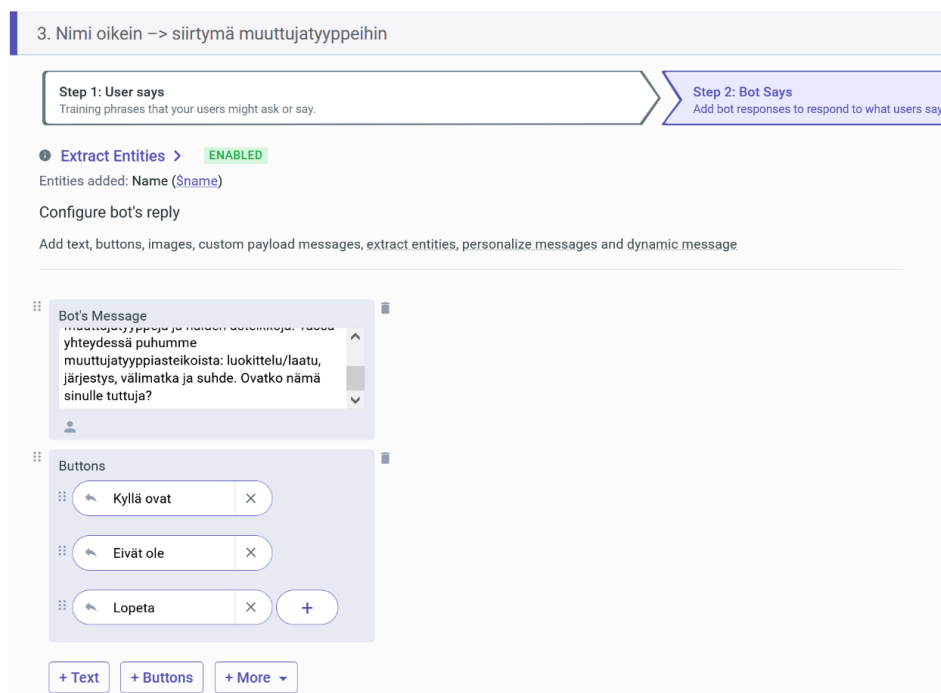
Kompose Chatbot on graafisen käyttöliittymän omaava platform-alusta, jolla agenttien luominen onnistuu ilman koodaamista. Käyttöönotto on vaivatonta: Kompose Chatbot ei vaadi asentamista omalle koneelle, joskin se on mahdollista, vaan agenteja pääsee luomaan suoraan selaimessa esimerkiksi Google-tilillä. Kieleksi voi valita sekä suomen että englannin ja lisäksi monia muitakin kieliä. Graafinen käyttöliittymä on aloittelijaystävällinen ja selkeä, ja se tarjoaa linkkejä dokumentaatioon ja ohjevideoihin, joilla agentin luomisessa pääsee nopeasti alkuun.

Kompose Chatbotista on saatavilla kolme erilaista tilausvaihtoehtoa: Start, Grow ja Business. Ensiksi mainitun hinta on 40 \$/kk ja Grow'n 100 \$/kk. Business-tilauksen hintaa ei kerrota. Vuositilauks on mahdollinen, jolloin saa kaksi kuukautta kaupan päälle. Start- ja Grow-tilauksia kaikkine ominaisuuksineen voi kokeilla ilmaiseksi 30 päivän ajan. Kallein Business-versio ei ole testattavissa ilmaiseksi. (Kommunicate, 2023b)

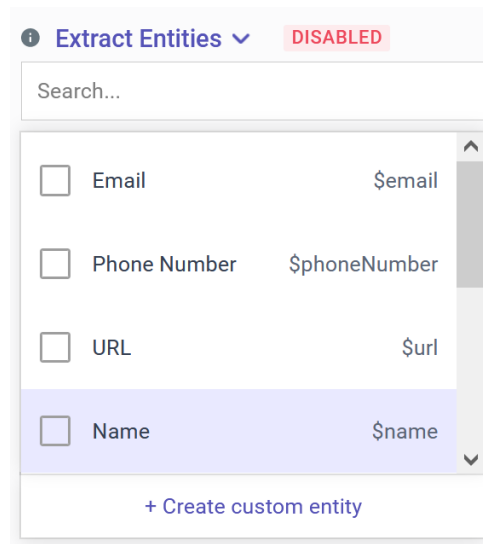
5.1.1 Kouluttaminen

Käyttöönoton lisäksi kouluttaminenkin on yksinkertaista selkeän graafisen käyttöliittymän ansiosta. Logiikan oppiminen ei ole lopulta vaikeaa, mutta vasta-alkaja saa silti tutkia ja kokeilla asioita jonkin verran, mikäli tarkoituksena on luoda monimutkaisempi agentti.

Kuvassa 17 näkyy hyvä yleiskuva Komposen toimintalogiikasta ja käyttöliittymästä: Welcome Message on nimensä mukaisesti tervetuloviesti, joka ilmestyy ruudulle heti agentin käynnistyessä. Sen alapuolella sijaitsevat Answer-lohkot ovat Kompose Chatbotin käytön ja kouluttamisen kannalta merkittävimpiä, sillä niillä onnistuu agentin toimintalogiikan luominen melko yksinkertaisesti. Vaihtoehtoina ovat muun muassa tekstimuotoinen viesti, kuva ja painike. Answer-lohkoja muokatessa piirtyy automaattisesti ja samanaikaisesti flow-kaavio (kuva 20).



Kuva 18 Kompose Chatbot - Toimintalogiikka



Kuva 19 Kompose Chatbot - Entiteetit

Ongelmia tuli vastaan siinä, että agentin valikoissa todetaan agentin osaavan poimia entiteettejä – nimiä, puhelinnumeroita, sähköpostiosoitteita ja niin edelleen – käyttäjien viesteistä, mutta esimerkiksi käyttäjän nimeä agentti ei silti osannut poimia talteen testausvaiheen aikana. Tämä entiteettien poimiminen näkyy kuvassa 18 kohdassa ”Extract Entities”. Kuvassa 19 näkyy myös muita entiteettejä, jotka Kompose Chatbot osaa poimia talteen käyttäjien syötteistä.

Kouluttamisen aikana ilmeni outoja synkronointiongelmia: PedAg-projektiryhmän yhteisellä tunnuksella luotu agentti ei testattaessa toiminut samoin joka tilanteessa saman PedAg-tunnuksen käyttämisestä huolimatta. Osalla testaajista agentti ei antanut ollenkaan ruudulle aloitus-/tervetuloviestiä. Edes selaimen vaihto ei auttanut tähän ongelmaan. Ongelmaa ei onnistuttu testauksen aikana ratkaisemaan, ja tämä onkin selkeä ongelma Kompose Chatbotissa.

Kompose Chatbotin automaattisesti luomassa flow-kaaviossa (kuva 20) vaikuttaa myös olevan ongelma, sillä se piirtyy loogisesti väärin, vaikka agenttia on koulutettu ohjeiden mukaan. Tähänkään ei löytynyt testauksen aikana syytä tai ratkaisua.



Kuva 20 Kompose Chatbot - Flow-kaavio

5.1.2 Integraatio

Kompose Chatbotin (2023) kotisivujen dokumentaatiossa mainitaan integraatiovaihtoehtoiksi esimerkiksi Moodle, WhatsApp, Line, Telegram, Dialogflow, IBM Watson ja ChatGPT. Mitään näistä ei kuitenkaan rajallisen testausajan vuoksi testattu. ChatGPT:n testaus ei ole edes mahdollista ilmaisella kokeiluversiolla.

5.2 Yhteenveto Kompose Chatbotista

Kompose Chatbot on yksinkertaisen graafisen käyttöliittymän ansiosta helppo ottaa käyttöön. Myös kouluttaminen on suhteellisen helppoa pienen oppimiskynnyksen jälkeen. Agentin luominen ilman koodaamista helpottaa agentin luomista. Ongelmiakin kuitenkin on, kuten flow-kaavion piirtyminen väärin ja synkronointiongelmia käyttäjien kesken. 30 päivän ilmainen kokeilu on riittävän pitkä aika testaamiseen, mutta toisaalta ilmaiseksi ei saa testata kaikkia ominaisuuksia, kuten tällä hetkellä pinnalla olevaa ChatGPT-

integraatiota. Kompose Chatbotin ohjevideoilla (Kommunicate, 2023c) pääsee alkuun, mutta edistyneempiä ominaisuuksia niissä ei kuitenkaan käsitellä.

6 Yhteenveto

Testatusta neljästä alustasta kolme on platform-alustoja: Botpress, Juji sekä Kompose Chatbot ja yksi framework-alusta Rasa. Tämä aiheuttaa jo eroja alustojen välille esimerkiksi käyttöönoton suhteen, sillä platform-alustojen käyttöönotto selaimessa ja käytettävyys graafisien käyttöliittymien vuoksi on helpompaa kuin komentotulkilta ladattava ja koulutettava Rasa. Paljon yhdistäviä tekijöitäkin kuitenkin löytyy. Kaikki testaukseen valitut alustat pystyvät hyödyntämään tekoälyominaisuuksia, vaikka eroja teknologioiden välillä on. Rasa on testatuista alustoista ainut, jonka kouluttamiseen tarvitaan koodin kirjoittamista. Kaikilla testatuilla alustoilla agentin voi kouluttaa kommunikoimaan millä tahansa luonnollisella kielellä, myös suomeksi. Rasa hyödyntää NLU:ta, NLP:tä ja NLG:tä. Botpressissä on käytössä kehittyneet NLP-mallit, NLU sekä GPT ja LLM. Juji hyödyntää NLU:ta ja Kompose Chatbot taas NLP:tä.

Kaikkia neljää alustaa päästiin testaamaan ilmaiseksi, joka oli testausvaiheeseen valikoitumisessa tärkeä kriteeri. Rasa on alustoista ainoa, jota pystyy käyttämään täysin ilmaiseksi. Jujilla agentin koulutus on ilmaista, mutta julkaisun jälkeen käyttö muuttuu maksulliseksi ja sitä kautta tarkempi testaus on haastavaa. Kompose Chatbotissa ilmainen kokeilujakso on suhteellisen lyhyt. Botpress on neljästä testatusta alustasta selkeästi kulurakenteeltaan edullisin, vaikka todellisessa käytössä tarvittavia viestimääriä voi olla hyvin haastavaa ainakaan aluksi arvioida.

Käyttötapausten kouluttamisella haluttiin testata, mitkä ominaisuudet alustalla onnistutaan agentille saamaan. Kaikilla alustoilla agentin syötteenä onnistui teksti, painikkeet, kuva sekä linkki. Aineiston syöttäminen agentilla saatiin onnistuneesti toimimaan vain Botpressillä, mutta myös Kompose Chatbotilla mahdollisuus aineiston syöttämiseen on. Käyttäjän syötteenä vastaus painikkeilla sekä avoimen vastauksen kirjoitus onnistuivat Kompose Chatbottia lukuun ottamatta kaikilla muilla testatuilla alustoilla.

Testatuista alustoista Botpress oli ainoa, jolla käyttötapaus saatiin koulutettua kaikilta osin onnistuneesti. Botpress oli myös ainoa alusta, jonka integraatiomahdollisuuksia päästiin kunnolla testaamaan. Näiden seikkojen ja myös alustan ilmaisen testauksen ja hinnoittelun edullisuuden vuoksi valikoitui Botpress pilotoitavaksi alustaksi projektin toiseen vaiheeseen.

Lähteet

Botpress. (2023a). Features. Haettu 26.3.2023 osoitteesta: <https://botpress.com/docs/cloud/getting-started/features/>

Botpress. (2023b). Botpress Cloud: Messaging Channels. Haettu 27.3.2023 osoitteesta: <https://botpress.com/docs/cloud/channels/overview/>

Botpress. (2023c). Next-generation chatbots that speak for themselves. Haettu 26.3.2023 osoitteesta: <https://botpress.com/features/gpt-native-engine>

Botpress. (2023d). Privacy Policy. Haettu 4.4.2023 osoitteesta: <https://botpress.com/privacy>

Botpress. (2023e). Pricing. Haettu 4.4.2023 osoitteesta: <https://botpress.com/pricing>

Botpress. (2023f). API Reference. Haettu 4.4.2023 osoitteesta: <https://botpress.com/docs/api/>

Botpress. (2023g). Languages. Haettu 9.4.2023 osoitteesta: <https://botpress.com/docs/cloud/getting-started/languages/>

Botpress Community. (2023). Unveiling the new Botpress: a GPT-native bot-building platform. Haettu 26.3.2023 osoitteesta: <https://botpress.com/blog/unveiling-the-new-botpress-a-gpt-native-bot-building-platform>

Businesswire. (2023). Botpress Launches Its New GPT-Native Platform For Building Chatbots That Can Execute Complex Workflows. Haettu 4.4.2023 osoitteesta: <https://www.businesswire.com/news/home/20230321005883/en/Botpress-Launches-Its-New-GPT-Native-Platform-For-Building-Chatbots-That-Can-Execute-Complex-Workflows>

Jira. (2023). Rasa Open Source, ohjelmistoprojekti. Haettu 27.3.2023 osoitteesta: <https://rasa-open-source.atlassian.net/jira/software/c/projects/OSS/issues>

Juji. (2022a). Introduction: The Juji Platform. Haettu 4.4.2023 osoitteesta: <https://juji.io/docs/>

Juji. (2022b). Juji Privacy Policy. Haettu 4.4.2023 osoitteesta: <https://juji.io/util/juji-privacy-policy.pdf>

Juji. (2022c). Juji API. Haettu osoitteesta 8.4.2023 osoitteesta: <https://juji.io/ai-chatbot-api/#how-chatbot-api-works>

Kommunicate. (2023a). Integrate your bot with Kommunicate. Haettu 5.4.2023 osoitteesta <https://docs.kommunicate.io/docs/bot-setup>

Kommunicate. (2023b). Pricing. Haettu 5.4.2023 osoitteesta: <https://www.kommunicate.io/pricing>

17.4.2023

Kommunicate. (2023c). Kommunicate: AI Chatbots and Support. Haettu 5.4.2023 osoitteesta: https://www.youtube.com/channel/UCUZbGHJ7kOA0A2bY_OFkGHQ

Nichol Alan. (2017). A New Approach to Conversational Software. Rasa Blog. Haettu 27.3.2023 osoitteesta: <https://medium.com/rasa-blog/a-new-approach-to-conversational-software-2e64a5d05f2a>

Rasa. (2023a). Rasa: About. Haettu 27.3.2023 osoitteesta: <https://www.linkedin.com/company/rasa/about/>

Rasa. (2023b). Introduction to Rasa Open Source & Rasa Pro. Haettu 27.3.2023 osoitteesta: <https://rasa.com/docs/rasa/>

Rasa. (2023c). Rasa Learning Centre. Haettu 27.3.2023 osoitteesta: <https://learning.rasa.com/>

Rasa. (2023d). Rasa Installation. Installing Rasa Open Source. Haettu 3.4.2023 osoitteesta: <https://rasa.com/docs/rasa/installation/installing-rasa-open-source>

Rasa. (2023e). Rasa Building Assistants. Command Line Interface. Haettu 4.4.2023 osoitteesta: <https://rasa.com/docs/rasa/command-line-interface/>

Rasa. (2023f). Conversation-Driven development. Haettu 7.4.2023 osoitteesta: <https://rasa.com/docs/rasa/conversation-driven-development>

Rasa. (2023g). NLG servers. Haettu 7.4.2023 osoitteesta: <https://rasa.com/docs/rasa/nlg/>

Rasa. (2023h). Tracker Stores. Haettu 7.4.2023 osoitteesta: <https://rasa.com/docs/rasa/tracker-stores/>

RasaHQ. (2021). RasaHQ/rasa-3.x-form-examples. Haettu 4.4.2023 osoitteesta: <https://github.com/RasaHQ/rasa-3.x-form-examples/blob/main/04-from-text/data/stories.yml>

RasaHQ. (2022). RasaHQ/helm-charts. Haettu 7.4.2023 osoitteesta: <https://github.com/RasaHQ/helm-charts/tree/main/charts/rasa/templates>

RasaHQ. (2023a). RasaHQ/rasa. Haettu 27.3.2023 osoitteesta: <https://github.com/RasaHQ/rasa>

RasaHQ. (2023b). Rasa. Haettu 27.3.2023 osoitteesta: <https://www.youtube.com/@RasaHQ>

RasaHQ. (2023c). Conversational AI with Rasa: Creating a New Assistant. Haettu 4.4.2023 osoitteesta: <https://youtu.be/IWP1K-lcAN4>

