

Slack-botti päivystäjätiedon välineenä



Ammattikorkeakoulututkinnon opinnäytetyö

Hämeenlinnan korkeakoulukeskus, Tietojenkäsittelyn koulutusohjelma

Kevät, 2021

Eemeli Sipiläinen

Koulutus

Kampus

Tekijä

Eemeli Sipiläinen

Vuosi 2020

Työn nimi

Slack-botti päivystäjätiedon välittäjänä

Työn ohjaaja /t

Lauri Salminen

TIIVISTELMÄ

Tämän opinnäytetyön tarkoituksena oli luoda Slack-botti Ambientia Group Oyn verkkotoimintojen keskuksen ja asiakaspalvelun päivystäjille. Päivystäjätietoa ylläpidetään Ambientian Intrassa (Confluence), joten tiedon hakeminen on autentikoinnin ja muutaman klikkauksen takana. Ratkaisuksi tähän tiedonhakuongelmaan ajateltiin Slack-botti, joka kertoisi tuon tiedon.

Botti tässä tarkoittaa tietokoneohjelmaa, joka toimii sille määriteltyjen toimintaohjeiden mukaan. Slack on ryhmätyöohjelmisto. Se on yritysten käyttöön tarkoitettu keskustelupohjainen viestintäsovellus. Sen tarkoituksena on korvata sähköposti ja auttaa ryhmää työskentelemään tehokkaammin.

Työntuloksena on Slack-botti, joka kehitettiin Node.js ja Slack bolt kehystä hyödyntämällä. Slack-botti hakee asiakaspalvelun ja ylläpidon puolella olevia päivystäjävoroja Bitbucketista ja näyttää niitä Slack keskusteluissa. Botin tekeminen onnistui hyvin ja se toimii niin kuin oli tarkoitettu. Kehittämiskyselyn kautta saadut arviot botista olivat myönteisiä.

Opinnäytetyön käytännön osuus osoitti, että Slack-botti on hyödyllinen työkalu erilaisten tehtävien automatisoinnissa, tässä tapauksessa päivystäjätietojen haun automatisoinnissa. Lisäksi osoitettiin, että Slack -alusta toimii hyvin botin rakentamisessa ja kehittämisessä. Teoriaosuudessa tuli esille, että erityisesti vuorovaikutteisten viesti- ja bottijärjestelmien suosio kasvaa jatkuvasti ja myös ohjelmistobottien käytön odotetaan lisääntyvän ja johtavan parannuksiin kehittäjien ja tiimien tuottavuudessa ja tehokkuudessa.

Avainsanat Node.js, Slack-botti, Keskustelubotti, Käyttöliittymä, Slack Bolt

Sivut 46 sivua, joista liitteitä 3 sivua

Sanasto

| | |
|-----|-----------------------------------|
| NOC | Network Operations Centre |
| API | Application Programming Interface |
| DM | Direct Message |

Name of degree programme

Campus

Author

Eemeli Sipiläinen

Year 2020

Subject

Slack bot as a tool for on-call information

Supervisors

Lauri Salminen

ABSTRACT

The purpose of this thesis was to create a Slack-bot for Ambientia Group Oy's "Online operations center" and "Customer service desk" - officers. Emergency information is maintained on Ambientia's Intranet (Confluence), so retrieving the information is behind authentication and a few clicks. The solution to this problem was thought to be a Slack-bot that would share that information.

A bot here refers to a computer program that operates according to the instructions assigned to it. Slack is a teamwork software. It is a chat-based messaging application for business use. Its purpose is to replace email and help the team work more efficiently.

The result is a Slack-bot developed using Node.js and Slack bolt. The Slack-bot searches for Bitbucket on-call customer service and maintenance and displays them in Slack chat. The bot was well made, and it works as intended. The bot estimates received through the development survey were positive.

The practical part of the thesis showed that Slack-bot is a useful tool for automating various tasks, in this case automating the retrieval of on-call information. In addition, the Slack platform was shown to perform well in bot construction and development. In the theory section, it has emerged that interactive messaging and bot systems in particular are gaining in popularity and that the use of software bots is also expected to increase and lead to improvements in the productivity and efficiency of developers and teams.

Keywords Node.js, Slack-bot, Chat-bot, User interface, Slack Bolt

Pages 46 pages including appendices 3 pages

SISÄLLYS

| | | |
|-------|---|----|
| 1 | JOHDANTO..... | 1 |
| 2 | TEHTÄVIEN AUTOMATISOINTI..... | 3 |
| 2.1 | Automatisoinnin tarkoitus | 3 |
| 2.2 | Älykkäät järjestelmät..... | 3 |
| 2.3 | Viesti- ja bottijärjestelmät..... | 5 |
| 2.4 | Botin määritelmä..... | 6 |
| 2.5 | Bottien roolit | 7 |
| 2.5.1 | Koodibotti | 8 |
| 2.5.2 | Testibotti..... | 8 |
| 2.5.3 | Kehitysoperaatiobotti..... | 8 |
| 2.5.4 | Tukibotti..... | 8 |
| 2.5.5 | Dokumentaatiobotti | 9 |
| 2.6 | Keskustelubotti tyypit | 9 |
| 2.6.1 | Valikko- / painikepohjaiset keskustelubotit | 9 |
| 2.6.2 | Avainsanatunnistukseen perustuvat keskustelubotit | 11 |
| 2.6.3 | Kontekstuaaliset keskustelubotit | 12 |
| 3 | SLACK JA SLACK-BOTTI..... | 13 |
| 3.1 | Slack-työtilat ja -keskustelut | 13 |
| 3.2 | Slack sovellukset..... | 13 |
| 3.3 | Slack-botti..... | 14 |
| 3.4 | Slack APIt | 14 |
| 3.4.1 | RTM API | 15 |
| 3.4.2 | Events API | 15 |
| 3.4.3 | Web API | 15 |
| 3.4.4 | Conversation API | 15 |
| 4 | TYÖKALUT JA TEKNOLOGIAT | 17 |
| 4.1 | Node.js..... | 17 |

| | | |
|-------|---|----|
| 4.2 | OAuth 2.0 | 17 |
| 4.3 | Bolt | 19 |
| 4.4 | Yaml..... | 19 |
| 5 | MENETELMÄT | 21 |
| 5.1 | Tiedon käsittely | 21 |
| 5.2 | Applikaation vaaditut Slack ominaisuudet ja toiminnallisuus | 23 |
| 5.3 | Interaktiivisen käyttöliittymän toteutus | 26 |
| 5.3.1 | Ylläpidon päivystäjäkäyttöliittymä | 26 |
| 5.3.2 | Asiakaspalvelun päivystäjäkäyttöliittymä | 32 |
| 5.3.3 | Muu käyttöliittymä | 35 |
| 5.4 | Botin käytettävyydestaus | 36 |
| 5.4.1 | Kyselyn tulokset..... | 37 |
| 6 | JOHTOPÄÄTÖKSET | 39 |
| | LÄHTEET | 41 |

Liitteet

Liite 1 Duty officer Slack Bot- survey

1 JOHDANTO

Tämä opinnäytetyö on tyypiltään toiminnallinen tutkimus keskustelubottiin käytöstä Ambientia -yrityksessä. Tutkimustyöhön liittyi käytännön suunnittelu ja koodausosuus, jossa rakennettiin keskustelubotti yrityksen käyttöön helpottamaan yhteydenottoa ja vuorovaikutusta työvuorossa olevaan päivystävään henkilökuntaan. Työssä on kyse rutiininomaisen tehtävän automatisoinnista ja toimijoiden välisen viestinnän helpottamisesta.

Työni toimeksiantajana toimii Ambientia Group Oy. Ambientia on keski-suuri yritys, joka tarjoaa asiakkailleen muun muassa palvelumuotoilua, ohjelmistokehitystä, pilvipalveluita, verkkokauppaa, IOT -palveluita, tekoälyä, ALM eli sovelluksen elinkaarihallintaa ja ITSM eli IT palveluhallintaa. Heillä on toimintaa Virossa ja Suomessa. Päätoimipaikka on Hämeenlinnassa. Toimistoja on myös Tampereella, Helsingissä, Joensuussa, Turussa, Tallinnassa ja Tartossa. (Ambientia n.d.)

Ambientialla on sekä asiakaspalvelussa että ylläpidon puolella kiertävä päivystäjävuoro. Päivystäjä ottaa vastaan erilaisia ongelmatilanteita ja auttaa niiden ratkaisemisessa tai tarpeen vaatiessa delegoi tehtäviä eteenpäin. Tällä hetkellä päivystäjätietoa ylläpidetään Ambientian Intrasta (Confluence), josta tiedon hakeminen on autentikoinnin ja muutaman klikkauksen takana. Ratkaisuksi tähän ongelmaan ajateltiin Slack-bottia, joka kertoisi päivystäjätiedot nopeasti ja varmasti sitä haluavalle.

Tätä varten päivystäjätietojen hallinta olisi jatkossa yaml-muotoisena Bitbucketissa, josta se on helppo koneellisesti hakea. Bottia varten pitäisi toteuttaa sovellus, joka tarjoaa (rest)-rajapinnan päivystäjätietojen hakemiseen.

Työssä etsitään vastauksia seuraaviin kysymyksiin:

- Kuinka Slack-bottia voidaan hyödyntää yrityksessä?
- Kuinka Slackiin voidaan tehdä integraatioita?
- Kuinka Slack-botti tehdään Node.jsllä?
- Kuinka Slack-botin käyttöliittymästä saadaan helppokäyttöinen?

2 TEHTÄVIEN AUTOMATISOINTI

2.1 Automatisoinnin tarkoitus

Automatisoinnin tarkoitus on auttaa kehittäjiä tekemään järkevämpiä päätöksiä ja tukemaan heidän keskinäistä vuorovaikutustaan. Ne ovat sovelluksia, jotka automatisoivat toistuvia tai ennalta määriteltyjä tehtäviä. Ohjelmistobottien käytön odotetaan lisääntyvän ja johtavan parannuksiin kehittäjien ja tiimien tuottavuudessa ja tehokkuudessa. (Storey & Zagalsky, 2016 s. 1)

Automatisointi ja digitalisaatio suuntautuvat yhä enemmän ihmisiin. Mitä onnistuneempi tekninen toteutus digitaalisella sovelluksella on, sitä inhimillisimmiltä ne vaikuttavat. Parhaat digitaaliset ratkaisut eivät hidasta meitä, vaan sen sijaan auttavat meitä saavuttamaan tavoitteitamme. Automatisoidut koneet saattavat kyetä tekemään uskomattomia asioita, mutta merkittävä ominaisuus menestyvissä tietoteknisissä järjestelmissä on se, että ne asettavat inhimillisen kokemussuunnittelun keskiöön. (Malcolm, Pring & Roehrig, 2017, s. 53)

2.2 Älykkäät järjestelmät

Älykkyyssjärjestelmä yhdistää ohjelmistot (algoritmit, liikesäännöt, koneoppimiskoodit, ennustavan analytiikan), laitteistot (palvelimet, anturit, mobiililaitteet, liitettävyys), datan (kontekstuaalinen ja reaaliaikainen) ja ihmisen antaman panoksen (usein arviointi tai kysymykset). (Malcolm, Pring & Roehrig, 2017, s. 46)

Artificial Intelligence (AI)– termin yksinkertainen määritelmä voi olla seuraava: ”AI on tietotekniikan alue, joka keskittyy koneisiin, jotka oppivat” (Malcolm, Pring & Roehrig, 2017, s. 48).

Kysee ei siis ole robotista, joka matkii ihmisen muotoa tai käyttäytymistä. Sen sijaan tekoäly edustaa seuraavan sukupolven tietokone- ja tietojärjestelmiä, jotka kuten vanhatkin tietokoneet ja järjestelmät olivat sijoitettuna ilmastoituihin tietokonehuoneisiin ja yhteydet niihin muodostettiin verkkojen kautta. (Malcolm, Pring & Roehrig, 2017, s. 48)

Tekoäly voidaan jakaa kolmeen alajoukkoon:

1. Suppea tekoäly (Narrow AI)
2. Yleinen tekoäly (General AI)
3. Super tekoäly (Super AI)

Suppea tekoäly voi olla nimeltään myös ”soveltuva tekoäly” tai ”heikko tekoäly”. Heikko tekoäly on tämän päivän koneiden älykkyyttä ja on vielä seuraavankin vuosikymmenen älykkyyttä, koska ne keskittyvät tosiasiallisesti vain yhden tietyn asian tekemiseen erittäin hyvin. Sellaisina nämä ”artificial narrow intelligence” tai ANI-järjestelmät olisivat toivottomia kaikissa muissa tehtävissä kuin mihin ne on erityisesti suunniteltu. Tällainen tekoäly on tiettyyn liiketoimintaan tai tehtävään, tuotteeseen tai palveluun keskittynyttä toimintaa, kuten auton ajaminen, röntgenkuvan lukeminen tai taloudellisten tapahtumien seuraaminen. (Malcolm, Pring & Roehrig, 2017, s. 49)

Yleinen tekoäly voi olla nimeltään myös ”vahva tekoäly” tai AGI. Tällöin koneella on samanlainen älykkyys kuin ihmisellä. Tällöin kone voi mennä kenen tahansa henkilön kotiin, jossa se myös osaa keittää kahvia, tunnistaa kahvinkeitin ja päätellä kuinka sillä tehdään kahvia ja kuinka painikkeet toimivat ja etsiä ja löytää kaapista kahvin jne. (Malcolm, Pring & Roehrig, 2017, s. 49)

Super tekoälyä Malcolm, Pring ja Roehrig (2017, s. 50) kuvaavat tekniseksi neroksi, joka on päästetty pullosta ja jota ihmiset eivät ehkä osaa pysäyttää. Heidän mukaansa emme osaa pysäyttää sellaista konetta tai

älykkyyttä, joka on meitä 10 tai 1000 askelta edellä. He toteavat, että tällaista älykkyyttä koneelta voi tuskin odottaa seuraavaan 100 vuoteen. Heidän mukaansa on turha murehtia superälykkäistä koneista, koska se on sama kuin murehtisi Marsin liikakansoittuvan jo ennen kuin olemme astuneet askeltakaan sen pinnalla.

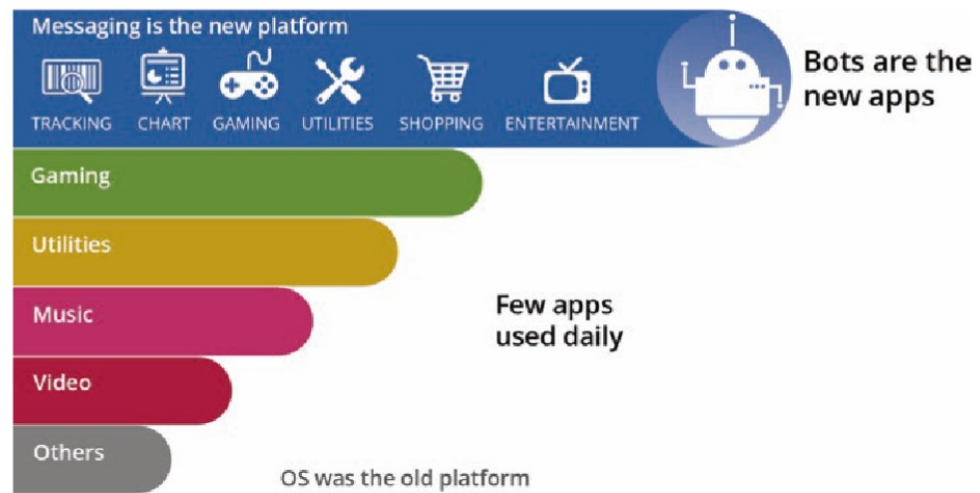
2.3 Viesti- ja bottijärjestelmät

Viesti- ja bottijärjestelmien suosio kasvaa jatkuvasti. Vuodesta 2015 lähtien ihmiset ovat käyttäneet merkittävästi enemmän sovelluksia eli applikaatioita kommunikaatioon kuin muita sosiaalisia verkkoja. Suosittuja sovelluksia ovat WhatsApp ja Facebook ja Aasiassa WeChat. Facebook ja Microsoft ovat ilmoittaneet keskittyvänsä botteihin. Microsoftin toimitusjohtaja Satya Nadella kuvaa botteja ”seuraavaksi suureksi asiaksi” (next big thing). Robotit ja Keskustelu alustat ovat Microsoftin strategian painopistealueena. (Gentsch, 2018, s. 82)

Microsoftin bottikehys luo olosuhteet bottien kehittämiseen eri alustoille tai omalle verkkosivustolle. Botin rakennuksen ohjelmistokehityksen työkalut (SDK) mahdollistavat bottien toteuttamisen. Kielen ymmärtämisen palvelu ”Language Understand Intelligence”(LUIS) auttaa bottia syväoppimisessä ja kielellisessä analytiikassa. Botit voidaan integroida eri viestijärjestelmiin bottiliittimen avulla. Bottihakemisto helpottaa muiden bottien löytämistä ja jakelua. Muita esimerkkejä alustoista, jotka mahdollistavat robottirakennuksen ja integroinnin, ovat Slack, Telegram ja Alk. (Gentsch, 2018, s. 83)

Bottien kehittyminen johtaa täysin erilaisiin periaatteisiin viestinnässä ja vastaavissa rajapinnoissa. Botit korvaavat suurimman osan verkkosivustoista ja sovelluksista. Asiakkaan kannalta tapahtuma voi sisältää esimerkiksi tuotteen arvioinnin, valinnan sekä oston ja palvelun. Tyypillisesti kuluttaja on käyttänyt näitä toimintoja varten erilaisia sovelluksia tai

verkkosivustoja. Botit eräänlaisena käyttöjärjestelmänä yhdistää erilaiset tiedon ja vuorovaikutuksen muodot universaaliksi yhdeksi tapahtumaksi (kuva 1).



Kuva 1 Botit yhdistävät vuorovaikutuksen muodot (Gentsch, 2018, s. 84)

2.4 Botin määritelmä

Botti on mukautettu suomeen englannin sanasta bot, joka on lyhenne muoto robotista. Sanastokeskuksen mukaan robottiohjelma eli bottiohjelma eli botti tarkoittaa tietokoneohjelmaa, joka toimii sille määriteltyjen toimintaohjeiden mukaan varsinkin suurta työmäärää, toistoa, jatkuvaa päivystystä ym. vaativissa tehtävissä, esimerkiksi hakuohjelmissa. Bottiohjelma voi kuitenkin olla myös roskapostia lähettävä haittaohjelma. (Kielikello n.d.)

Sana botti saattaa tulla vastaan, kun Internet-sivun käyttäjän pitää vaikkapa keskusteluun osallistuakseen osoittaa (yleensä kuva- tai sanatestin avulla), ettei ole botti. (Kielikello n.d.)

Koska botti-sanan merkitys ei ole yksiselitteinen, kannattaa sen sijasta usein käyttää jotain muuta ilmausta. Esimerkiksi Kelassa on kokeiltu

verkkokeskustelua (= chat, tsätti), jossa osapuolena on keskustelurobotti. Sanan perusteella asiakas tietää, että keskustelukumppanina ei ole ihminen vaan kone. (Kielikello n.d.)

Perusmuodossaan botit toimivat kanavana tai käyttöliittymänä käyttäjien ja palveluiden välillä, tyyppillisesti keskustelukäyttäjäliittymän kautta. (Elliott, 2018, s. 119)

Botteja kehitetään edelleen lisäämällä niihin persoonallisuutta ja muistia. Ne voidaan suunnitella toimimaan ”vetotilassa”, jossa käyttäjä käynnistää vuorovaikutuksen ja ”työnnä tilassa”, jossa botti aloittaa vuorovaikutuksen, tai näiden yhdistelmänä. Yleisimmin botteja käytetään tehtävien automatisointiin (esim. testien suorittamiseen, kun ennalta määritellyt ehdot täyttyvät) tai sovellustyökalujen yhdistämiseksi toisiinsa. Botit voivat hyödyntää tekoäly- tai koneoppimistekniikoita, tai ne saattavat kaapata tai analysoida muiden työkalujen ja robotien tuottamaa tietoa. (Storey & Zagalsky, 2016 s. 1)

Tässä työssä käytetään sanaa botti, joka tietojenkäsittelystä puhuttaessa yleisesti ymmärretään tarkoittavan määriteltyjen ohjeiden mukaisesti käyttäytyvää tietokoneohjelmaa.

2.5 Bottien roolit

Tehtävien automatisoinnissa merkittävin kehityskulku on viestibottien esiintulo ja nousu. Tekoälyllä varustetut botit tai keskustelubotit edustavat Internet sovellusten uusinta sukupolvea. Keskustelubotti tekee tehtäviä tai toimii käyttäjän kanssa sovellusliittymän kautta joko kirjoittamalla tai puhumalla. Viesti tai vuorovaikutusbotteja on hämmästyttävä määrä erilaisia, joista osa on kykenevä automaattiseen keskusteluun tai toimittamaan monimuotoisia tehtäviä. (Elliott, 2018, s. 119)

Sisältöbotit (Content Bots) keräävät ja jakavat haluttua sisältöä käyttäjien kanssa, kuten urheilu-uutisia. Ruokabotit (Food Bots) tilaavat tai järjestävät ruokatilauksia. E – tilaus botit (e-Commerce Bots) ostavat tuotteita tai palveluita. Työnkulku botit (Workflow Bots) automatisoivat yritysten työnkuluja kuten liiketoimintaraportteja myyntiorganisaatiolle, operaatioita hallinnolle ja taloudelle. IOT-botit (Internet of Things) yhdistävät ihmiset heidän laitteisiinsa, autoihin ja älykoteihin. (Elliott, 2018, s.119)

2.5.1 Koodibotti

Koodirobotit ovat ohjelmistorobotteja, jotka voivat kirjoittaa merkittävän osan sovelluksen koodista ja siten auttavat tehostamaan koodityöskentelyä. Ne voivat myös auttaa koodimuutosten huomaamisessa, tekemisessä ja korjaamisessa. (Storey & Zagalsky, 2016, s. 2)

2.5.2 Testibotti

Testibotti, kuten FindBugs on ohjelma, joka käyttää staattista analyysia, jonka avulla se etsii virheitä Java-koodista. (Storey & Zagalsky, 2016, s.2)

2.5.3 Kehitysoperaatiobotti

Kehitysoperaatiobotti (DevOps) ilmoittaa ongelmista palvelussa ja vie siitä viestin ongelmanratkaisijalle. (Storey & Zagalsky, 2016, s. 2)

2.5.4 Tukibotti

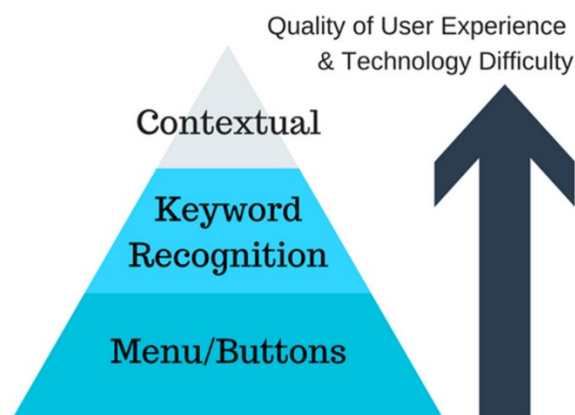
Tukibotit auttavat kehittäjien ja käyttäjien välisessä kommunikaatiossa. Botit voivat vastata automaattisesti usein kysytyihin kysymyksiin ja siten helpottaa esim. opettajien työtä oppilaiden kanssa. (Storey & Zagalsky, 2016, s.3)

2.5.5 Dokumentaatiobotti

Dokumentaatiobotit tekevät muistiinpanoja yhdistämällä tietoja koodista ja niihin liittyvistä kommentteista, joita voidaan sitten käyttää sovelluksen julkaisusta tiedottamisen yhteydessä. (Storey & Zagalsky, 2016, s.3)

2.6 Keskustelubotti tyypit

Kaikkia keskustelubotteja ei ole tehty samanlaisiksi. Keskustelubotteja on erimuotoisia, kokoisia ja kyvyiltään erilaisia. Peruskeskustelubotit riittävät moneen tehtävään, mutta jotkut tehtävät vaativat edistyneempiä keskustelubotteja. Keskustelubotti on hyvä tehdä tehtävien vaatimusten mukaiseksi. Joskus yksinkertainen klikattava valikkopohjainen botti riittää hoitamaan homman. (Phillips, 2018)



Kuva 2 Teknisen monimutkaisuuden kasvaessa laadukkaan käyttökokemuksen saanti vaikeutuu (Phillips, 2018)

2.6.1 Valikko- / painikepohjaiset keskustelubotit

Valikkopainotteiset keskustelurobotit ovat kaikista suosituimpia ja alkeellisimpia. Nämä koostuvat valintapuuhierarkiasta, jota käyttäjälle näytetään painikkeiden muodossa. Käyttäjä joutuu tekemään useita

painikevalintoja päästäkseen lopulliseen vastaukseen. Painikepohjaiset keskustelubotit ovat myös hitaampia kuin muut vaihtoehdot. (Phillips, 2018)

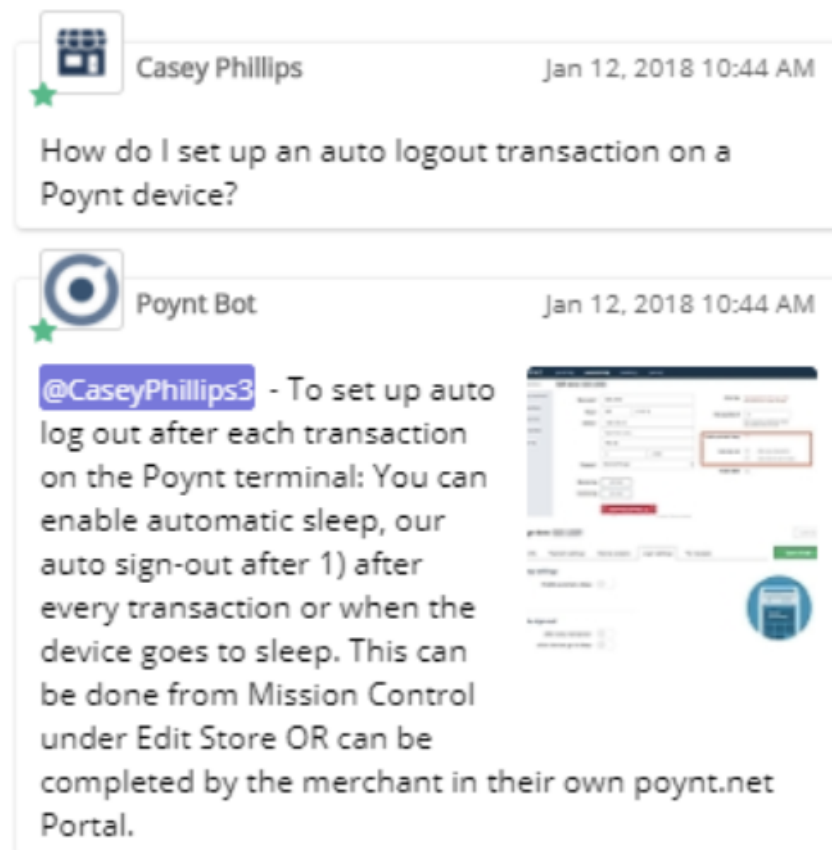


Kuva 3 Valikkopainikkeet ohjaavat käyttäjiä (Phillips, 2018)

2.6.2 Avainsanatunnistukseen perustuvat keskustelubotit

Avainsanatunnistukseen perustuvat keskustelubotit lukevat mitä käyttäjät kirjoittavat ja yrittävät vastata käyttäjän kysymyksiin tai etsivät tekstistä avainsanoja määrittääkseen kuinka palvella asiakkaita. Useat samankaltaiset kysymykset aiheuttavat vaikeuksia näille keskusteluboteille.

Jos käyttäjä esimerkiksi kysyy kysymyksen: ‘How do I set up an auto logout transaction on a Poynt device?’, Botti käyttää avainsanoja ‘auto’, ‘logout’, and ‘Poynt’, joilla se määrittelee miten vastaa kysymykseen.

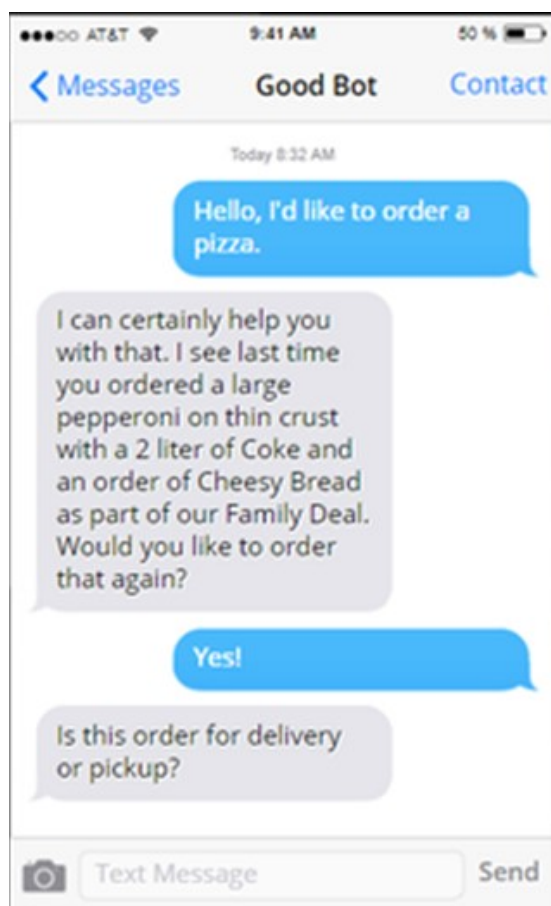


Kuva 4 Botti käyttää avainsanatunnistusta löytääkseen oikean vastauksen (Phillips. C, 2018)

Edellä esitetyn vuoksi on tullut suosituksi käyttää hybridikeskustelubotteja, jotka muodostuvat yhdistelmästä valikkopohjaista ja avainsanatunnistuspohjaista keskustelubottia. (Phillips, 2018)

2.6.3 Kontekstuaaliset keskustelubotit

Kehittyneimpiä botteja ovat kontekstuaaliset keskustelubotit. Niissä hyödynnetään tekoälyä sekä koneoppimista. Kontekstuaalinen keskustelubotti muistaa keskustelut tietyn käyttäjän kanssa ja oppii käymistään keskusteluista. Se tallettaa tiedon jokaisesta keskustelusta ja oppii tietämään mitä asiakas haluaa esim. tilata ja tietää mikä on asiakkaan osoite ja maksutiedot (kuva 5). (Phillips, 2018)



Kuva 5 Pizzan tilaaminen kontekstuaalisen keskustelubotin avulla. (Phillips, 2018)

3 SLACK JA SLACK-BOTTI

Slack on ryhmätyöohjelmisto. Se on yritysten käyttöön tarkoitettu keskustelupohjainen viestintäsovellus. Sen tarkoituksena on korvata sähköposti ja auttaa ryhmää työskentelemään tehokkaammin. (Slack n.d.)

3.1 Slack-työtilat ja -keskustelut

Slack-työtila koostuu kanavista. Kanavissa ihmiset voivat työskennellä yhdessä muiden kanavan jäsenten kanssa. Kanavat voivat olla suljettuja tai avoimia käyttöoikeuksien mukaan. Kanavissa on vierasprofiili ja jäsenprofiili. Kanavilla on myös omistajat, jotka luovat työtilan ja asettavat käyttöoikeudet. Omistajat nimeävät järjestelmänvalvojat auttamaan ryhmää. Jäsenet voivat lähettää viestejä ja tiedostoja ryhmän jäsenille. Kanavia voi perustaa tiimille, projektille, ulkoisille asiakkaille tai mille tahansa ryhmälle, joka nähdään tarkoituksenmukaiseksi. (Slack n.d.)

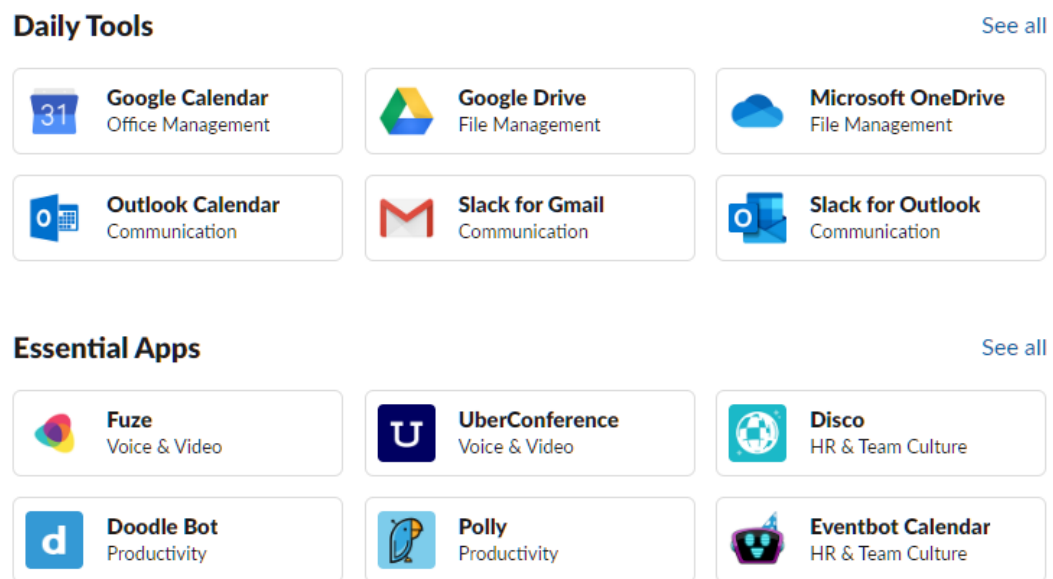
Slackissa lähetetään tiimin jäsenille viestejä ja siellä jaetaan tiedostoja. Keskusteluhistoria löytyy Slackista. Keskusteluista muodostuu yleistä yrityksen institutionaalista tietoa. Keskustelut voivat olla yksityisiä tai julkisia kaikille jäsenille. Keskusteluja voi käydä myös organisaation ulkopuolisten kanssa. (Slack n.d.)

Viestit ja tiedostot, jotka jaetaan julkisella kanavalla jäävät yrityksen julkiseksi tai tiimin historiatiedoksi. Tiimin keskustelut ja merkitykselliset tiedostot voi hakea esille tiimin historiatiedoista. (Slack n.d.)

3.2 Slack sovellukset

Slack-työtilaan voi asentaa erilaisia käyttökelpoisia sovelluksia ja työkaluja helpottamaan työtä. Tarjolla on tuhansia sovelluksia, joilla voi organisoida

ja tehostaa työskentelyä. Sovellukset löytyvät Slack App Directorystä.
(Slack n.d.)



Kuva 6 Sovelluksia Slack App Directorystä.

3.3 Slack-botti

Slack-botti on Slack sovellus, joka on vuorovaikutuksessa käyttäjänsä kanssa keskustelemalla. Botti on sama kuin normaali sovellus. Se voi käyttää samoja API alueita, kun sovellus. Kun applikaatiolle annetaan kasvot, nimi ja persoonallisuus siitä tulee Slack sovellus.

Slack-botit odottavat viestejä, joissa on vihjesanoja, joihin ne voivat vastata. Slack-botti voi myös lähettää suoria tiedostoviestejä, käyttäjät voivat mainita sen ja se voidaan kutsua kanaville tai potkaista pois. (Slack n.d.)

3.4 Slack API:t

Slack-alusta koostuu useista sovellu ohjelmointirajapinnoista, joiden avulla voit rakentaa sovelluksia. Yleensä nämä ohjelmointirajapinnat toimivat

yhdessä, mutta niillä on myös vähän päällekkäisyyksiä, jotka voivat joskus olla hämmentäviä. Jollakin ohjelmointirajapinnoilla on jonkin verran samanlaisia ominaisuuksia, mutta vaativat erilaisia tapoja yhdistää niihin. (Slack, n.d)

3.4.1 RTM API

The Real Time Messaging (RTM) API on tapa yhdistää toinen applikaatio Slackiin. RMT toimii websocketeilla, jotka pitävät yhteyden Slackin ja applikaation välillä. Kaikki aktiivisuus ja jokainen julkinen viesti sekä uuden kanavan luominen tapahtuu tässä työtilassa. Tämän takia liikenne voi olla runsasta ja ylläpitämisestä voi tulla työlästä. Tämän takia on suositeltavaa käyttää muita API vaihtoehtoja applikaation rakentamiseen. (Slack, n.d)

3.4.2 Events API

Events APIa on hyödyllistä mieltä evoluutiona RTM APIsta. Sen sijasta, että tehdään jatkuvasti aktiivinen yhteys, joka saa Slackin toimimaan, valitaan tapahtumia, joista ollaan kiinnostuneita ja Slack lähettää tämän http pyynnöllä tiettyyn päätepisteeseen (endpoint), jonka on määritetty erikseen. Events API voi toimittaa käyttäjälle tiedon esimerkiksi siitä, milloin henkilöitä liittyy kanavalle tai reagoi emojiin. (Slack, n.d)

3.4.3 Web API

Web APIlla tehdään toimintoja työtilassa kuten laaditaan viestejä tai muodostetaan uusia kanavia. Sillä tehdään myös suoria kyselypyyntöjä, kuten pyydetään listaus työtilassa olevista henkilöistä. (Slack, n.d)

3.4.4 Conversation API

Conversation APIssa on funktioita, joilla ohjataan Slackia suoraan. Nämä toiminnot käsittelevät erityisesti kaikkia kanaviin liittyviä asioita, joita on

Slack – työtilassa kuten julkiset ja yksityiset kanavat, jaetut kanavat, ulkoiset kanavat, yksityiset viestit ja ryhmäviestit. Suurimmaksi osaksi nämä kaikki kanavamaiset kohteet käyttäytyvät samalla tavalla (voit esimerkiksi luetteloida niillä olevia käyttäjiä). Jokaisella kanavalla on kuitenkin omat ainutlaatuiset ominaisuutensa, kuten kanavan yksityisyys tai julkisuus. (Slack, n.d)

4 TYÖKALUT JA TEKNOLOGIAT

Tässä luvussa käydään läpi työkalut ja teknologiat, joita käytetään botin rakentamisessa ja kehittämisessä.

4.1 Node.js

Node.js on avoimen lähdekoodin alusta, joka mahdollistaa skaalautuvien ja nopeiden internet -sovellusten rakentamisen JavaScriptillä. Node.js pohjautuu Google Chrome V8 JavaScript virtuaalikoneeseen, joka käyttää Google Chromen nettiselainta. (Teixeira, 2013, s. 11)

Node.js on kevyt ja tehokas, tapahtumapohjainen virtuaaliympäristö ja se pystyy käsittelemään useita samanaikaisia verkkoyhteyksiä, mikä tekee siitä käyttökelpoisen dataintensiivisissä sovelluksissa. (Teixeira, 2013, s. 11)

Node.js voi käyttää ulkopuolisten toimijoiden avoimen lähdekoodin moduuleja. Käyttämällä Node Package Manager (NPM) hallintaa, voidaan asentaa, hallita ja käyttää mitä tahansa niistä monista moduuleista, jotka sisältyvät alati kasvavaan moduulikirjastoon. (Teixeira, 2013, s. 12)

4.2 OAuth 2.0

”The OAuth 2.0 Authorization Framework” (OAuth 2.0) on tietyn palvelun suojakerros, jonka vuoksi palvelua varten rakennetuilla asiakassovelluksilla on standardoitu menetelmä pyytää järjestelmää käyttävälle palvelulle tai käyttäjälle kuuluvia suojattuja resursseja. Asiakassovellukset tekevät HTTP – pyyntöjä ennalta määritettyihin palvelun päätepisteisiin ja toimittavat samalla käyttöoikeustunnuksen, joka kertoo, mikä käyttäjä on hyväksynyt asiakassovelluksen pääsyn tietoihinsa. OAuth on valtuutusprotokolla, joka suojaaa yleisimmin REST sovellusliittymiä ja verkkopohjaisia sovelluksia.

REST on arkkitehtuurityyli, joka perustuu periaatteisiin, jotka kuvaavat kuinka verkkoon liitetyt resurssit määritellään ja käsitellään. (Spasovski, 2013, s. 23)

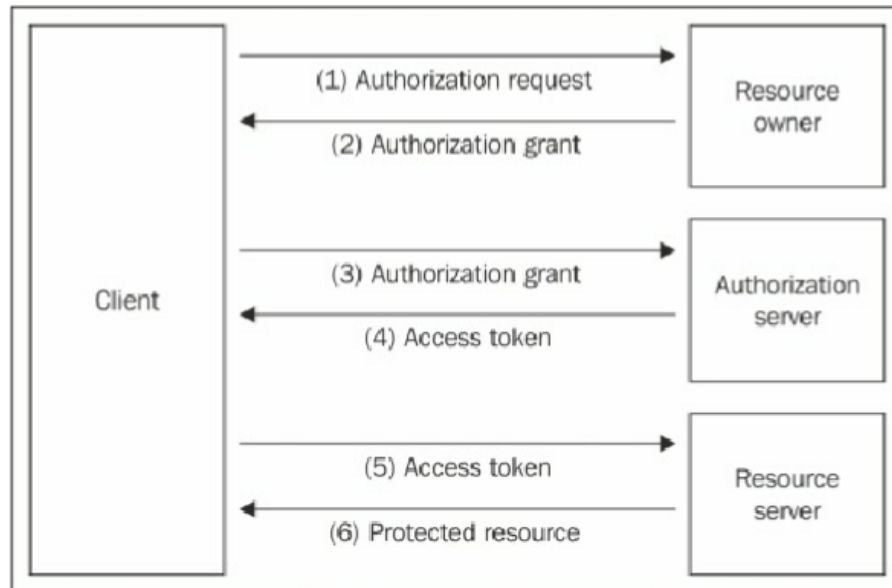
OAuth 2.0 suojakerrosta käyttävät muun muassa PayPal, Amazon, Facebook ja Twitter. Sitä käytetään erilaisissa tilanteissa palvelusovellusliittymien turvaamisesta aina helppokäyttöiseen kirjautumismekanismiin. (Spasovski, 2013, s. 23)

OAuth 2.0 kehystä voidaan tarkastella roolien kautta (kuva 7). Resurssien omistaja on yksikkö, jolla on oikeus myöntää oikeuksia suojattuihin resursseihin. Kun yksikkö on henkilö niin häntä voi sanoa käyttäjäksi tai loppukäyttäjäksi. (Spasovski, 2013, s. 30)

Käyttövaltuuspalvelin kysyy resurssien omistajalta (käyttäjältä) vahvistusta, että asiakkaalle voidaan myöntää käyttövaltuus. Palvelin antaa asiakkaalle käyttöoikeustunnuksen (access token). (Spasovski, 2013, s. 30)

Resurssipalvelin on palvelin, jossa ovat suojatut resurssit ja jolle asiakassovellus tekee valtuutettuja pyyntöjä. (Spasovski, 2013, s. 31)

Asiakas (client) on sovellus, joka pyytää resurssipalvelimelta suojattuja resursseja omistajan (käyttäjän) puolesta. Asiakassovellus voi olla verkkosovellus, työpöytäsovellus tai mobiilisovellus. (Spasovski, 2013, s. 31)



Kuva 7 OAuth 2.0 autentikointiprosessi

4.3 Bolt

Bolt on Slackin kehittämä JavaScript kehys, joka tekee Slack aplikaatioiden rakentamisen helpommaksi. Siinä on mukana alustan uusimmat ominaisuudet. Slack Bolt käsittelee merkittävän osan perusasetuksista kuten tunusten validoinnin ja palvelintuen. Boltissa on automaattinen tekstin täydennys ja Type Script – tuki, joka helpottaa koodieditorin käyttöä. Boltissa on yksinkertaisia käyttöliittymiä yleisimmille sovellusten ominaisuuksille kuten viestien lähettämiseksi, Web- puheluille ja tapahtumien kuuntelemiseksi ja vuorovaikutteisuuden lisäämiselle. (Slack n.d.)

4.4 Yaml

YAML ("Yet Another Markup Language") on Unicode-pohjainen tietojen merkintäkieli, joka on suunniteltu ketterien ohjelmointikielten yleisten natiivitietojen ympärille. YAMLia käytetään usein konfiguraatiodoistoissa, mutta sitä voi käyttää myös muunlaiseen tietojen talletukseen. Se on laajalti hyödyllinen ohjelmointitarpeissa, jotka vaihtelevat

kokoonpanotiedostoista Internet-viesteihin, objektien pysyvyydestä tietojen tarkastukseen. (yaml, n.d.)

YAML on suunniteltu alusta alkaen hyödylliseksi ja käyttäjäystävälliseksi datatietojen kanssa työskenteleville. Se käyttää Unicode-tulostettavia merkkejä, joista osa tarjoaa datan rakenteellista tietoa ja loput datan sisältötietoja. YAML minimoi rakenteellisten merkkien määrän, joka auttaa dataa näkymään merkityksellisellä tavalla. Esimerkiksi sisennystä voidaan käyttää rakenteessa, kaksoispisteet erottamaan avaimia: arvoparit ja viivat luodaan "luetteloihin". (Yaml, n.d.)

Tietorakenteissa on lukemattomia vaihtoehtoja, mutta ne kaikki voidaan esittää kolmella perusprimatiivilla: mappings (hashes/dictionaries), sequences (arrays/lists) and scalars (strings/numbers). YAML hyödyntää näitä perusmäärittelyjä ja lisää mukaan yksinkertaisen kirjoitusjärjestelmän ja laskostumismekanismen, josta muodostuu täydellinen kieli kaikkien natiivien tietorakenteiden sarjoittamiseksi. Vaikka suurin osa ohjelmointikielistä voi käyttää YAMLia datan sarjanumerointiin, YAML käyttäytyy erinomaisesti niiden kielten kanssa, jotka on rakennettu pohjimmiltaan kolmen perusprimaarin ympärille. Näitä ovat ketterät kielet, kuten Perl, Python, PHP, Ruby ja Javascript. (Yaml, n.d.)

YAML luotiin erityisesti toimimaan yleisissä käyttötapauksissa, kuten tiedostojen konfigurointi, lokitiedostojen luonti, viestien prosessointi, kieltenvälinen tietojen jakaminen, objektien pysyvyys ja monimutkaisten tietorakenteiden virheenkorjaus. Kun tietoja on helppo tarkastella ja ymmärtää, ohjelmoinnista tulee yksinkertaisempaa. (Yaml, n.d.)

5 MENETELMÄT

Käytännön osuudessa on tarkoituksena kehittää Slack keskustelubotti. Botti on kehitetty ja tarkoitettu Ambientian työntekijöille. Botille kehitetään komentoja, joilla päivystäjä tietojen hakeminen nopeutuu. Seuraavissa kappaleissa käydään läpi botin datan ja käyttöliittymän toiminnallisuuden rakentaminen.

5.1 Tiedon käsittely

Asiakkaan asiakaspalvelun sekä ylläpidon päivystäjätiedot, joita minä lähdin käsittelemään, sijaitsivat alun perin asiakkaan intranetissä. Tiedon hakeminen intranetistä vaatii autentikoinnin ja klikkauksia väliin. Tiedon haulle piti sen vuoksi keksiä parempi ratkaisu.

Asiakaspalvelun sekä ylläpidon päivystäjätiedot ovat intranetissä taulukkomuodossa. Taulukossa 1 on asiakaspalvelun päivystäjät ja taulukossa 2 on ylläpidon päivystäjät.

Rakensin päivystäjistä YAML-tilukot lajittelemalla jokaisen viikkotaulukon ja päivätilukon sarakkeet omaksi objektikseen, ja näissä ovat rivitiedot objektin arvona. Viikkosarake jaoteltiin vielä päivän numeron, viikon ja päivän nimen mukaan, tiedon määrän selvennyksen vuoksi.

Taulukko 1 kuvaa asiakaspalvelun päivystäjiä ja tästä tiedosta kuvastui kuva 8 muotoinen YAML tieto.

Taulukko 1 Asiakaspalvelun päivystäjät

| Week 6 | | Morning (8-12) | Afternoon (12-16) | Notice |
|--------|------|-------------------|----------------------|--------------|
| Mon | 3.2. | @jari.talo | @erkki.tina | |
| Tue | 4.2. | @hannu.hiden | @jari.kari | |
| Wed | 5.2. | @markku.vesi | @mari.kuru | |
| Thu | 6.2. | @essi.salo | @kalle.karu | |
| Fri | 7.2. | @anni.onni | @markku.vesi | @hanna.heino |

- notice:

date:

week: 6

date: 3.2.2020

day: Mon

morning: @jari.talo

afternoon: @erkki.tina

...

Kuva 8 Asiakaspalvelun päivystäjien maanantain tiedot YAML – muodossa

Taulukko 2 kuvaa asiakaspalvelun päivystäjiä ja tästä tiedosta kuvastui kuva 9 muotoinen YAML tieto

Taulukko 2 Ylläpidon päivystäjät

| Week / BU | Morning NOC (07-13) | Evening NOC(13-19) | Notes | 24x7 |
|-----------------------|------------------------|-----------------------|-----------------|--------------|
| Week 2 (7.1-10.1) | @ville.valo | @niko.salo | Loppi- ainen | @hirvi.hanhi |
| Week 3 (13.1-17.1) | @halo.gael | @kalle.wok | | @ville.valo |
| Week 4 (20.1-24.1) | @rafael.ranta | @essi.ranta | | @elli.kanta |

| | | | | |
|-----------------------|---------------|-------------|--|---------------|
| Week 5 (27.1-13.1) | @halo.gael | @jari.vesi | | @hirvi.hanhi |
| Week 6 (3.2-7.2) | @erkki.sandel | @kalle.wok | | @irma.jalo |
| Week 7 (10.2-14.2) | @niko.salo | @kalle.wok | | @halo.gael |
| Week 8 (17.2-21.2) | @irma.jalo | @ville.valo | | @essi.ranta |
| Week 9 (24.2-28.2) | @kalle.wok | @elli.kanta | | @erkki.sandel |

- notes: loppiainen

date:

week: 2

start: 13.1.2020

end: 17.1.2020

morningNoc: AES @ville.valo

eveningNoc: CS @niko.salo

24x7: "@hirvi.hanhi"

...

Kuva 9 Ylläpidon päivystäjien viikon 2 tiedot YAML – muodossa

5.2 Applikaation vaaditut Slack ominaisuudet ja toiminnallisuus

Botti vaati seuraavia toimintoja toimiakseen. Näitä hallitaan Slack API applikaatio sivun puolelta (<https://api.slack.com/apps>). Botti sovellus hyödyntää seuraavia toimintoja ja rajoja.

Slash-komennot antavat käyttäjille mahdollisuuden kutsua Slack-bottia kirjoittamalla viestin kirjoituskenttään. Lähetetty Slash-komento aiheuttaa hyötykuorman tietojen lähettämisen Slackista siihen liittyvään sovellukseen. Sovellus voi sitten vastata millä tahansa tavalla haluamallaan

hyötykuorman tarjoamalla kontekstilla. Nämä komennot ovat lähtökoh-
tana monimutkaisille työnkuluille, integroinneille ulkoisiin palveluihin tai
yksinkertaisiin viestivastauksiin.

Slash-komennon luominen tapahtuu Slackin ”app management dashboard”
valikosta. Slash-komento pyytää seuraavia parametreja, jotka olen määri-
tellyt taulukoissa. Taulukko 3 kuvaa ylläpidon päivystäjä komentoa ja tau-
lukko 4 asiakaspalvelun päivystäjä komentoa.

Taulukko 3 Ylläpidon Slack- komento

| Määrittäminen | parametrit | Kuvaus |
|------------------------|---|---|
| Command | /dutyofficernoc | komennon nimi, todellinen merk- kijono, jonka käyttäjät kirjoittavat aloittaakseen vuorovaikutuksen. |
| Request URL | https://dutyoffi- cer.fi/slack/com- mands | URL osoite, johon lähetetään hyö- tykuorma. Kun komento syöte- tään. |
| Short desc- ription | Displays NOC duty officers | Lyhyt kuvaus komennon toimin- nasta. |
| Usage Hint | week number | Tämä näkyy käyttäjille, kun he yrittävät kutsua komentoa. Jos ko- mennossa on parametrejä niitä voi syöttää komennon perään. |

Taulukko 4 Asiakaspalvelun Slack- komento

| Määrittäminen | parametrit | Kuvaus |
|---------------|----------------|--|
| Command | /dutyofficersd | komennon nimi, todellinen merk- kijono, jonka käyttäjät kirjoittavat aloittaakseen vuorovaikutuksen. |

| | | |
|-------------------|---|--|
| Request URL | https://dutyofficer.fi/slack/commands | URL osoite, johon lähetetään hyötykuorma. Kun komento syötetään. |
| Short description | Displays sd duty officers | Lyhyt kuvaus komennon toiminnasta. |

Interaktiiviset komponentit ovat osa Block Kit -elementtejä, jotka lisäävät interaktiivisuutta painikkeisiin. Painikkeen painalluksesta lähtee käyntiin vuorovaikutustapahtuma.

1. Käyttäjä käynnistää vuorovaikutuksen käyttämällä yhtä sovelluksen sisäntulopisteistä, esimerkiksi painiketta.
2. Sovellus vastaanottaa ja käsittelee vuorovaikutuksen hyötykuorman.
3. Tämän kontekstin avulla sovellus luo vastauksen vuorovaikutukseen.

Jokaisessa interaktiivisessa komponentissa on "action id", jolloin painikkeen painalluksessa painike tietää minne päätepisteeseen se lähettää http pyynnön ja jatkaa vuorovaikutuksen käsittelyä eteenpäin.

API luvat määritetään "Permission" valikosta. Tämä sivu määrittelee OAuth rajoja botille (taulukko 5).

Taulukko 5 Slack-botti OAuth rajat.

| OAuth raja | Kuvaus |
|------------|---|
| chat:write | Send messages as @duty-officer-info |
| Commands | Add actions and/or slash commands that people can use |

Näiden kaikkien ominaisuudesta ja toiminnallisuudesta rakentuu Slack-bottiin rajat. Kuten kuvasta 10 huomataan Slack-botti käyttää slash komentoja ja interaktiivisia komponentteja.

```
const app = new App({
  signingSecret: process.env.SLACK_SIGNING_SECRET,
  token: process.env.SLACK_BOT_TOKEN,
  endpoints: {
    commands: '/slack/commands',
    'interactive-endpoint': '/slack/interactive-endpoint'
  },
  logLevel: LogLevel.DEBUG
});
```

Kuva 10 Slack Bolt botit App

5.3 Interaktiivisen käyttöliittymän toteutus

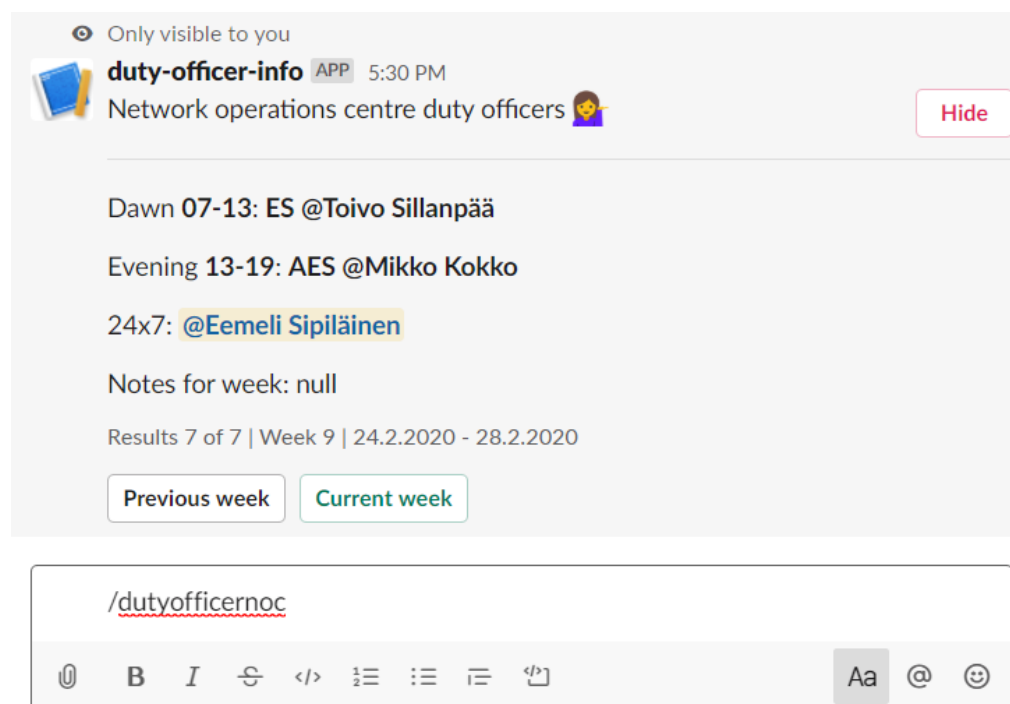
Työn seuraavana vaiheena on interaktiivisen käyttöliittymän luominen. Käyttöliittymän käytön käyttäjille tulisi olla intuitiivista.

5.3.1 Ylläpidon päivystäjäkäyttöliittymä

Ensimmäisenä käyttöliittymäkokonaisuutena on ylläpidon päivystäjien (network operations centre, NOC) hakujen toteuttaminen. NOC-päivystäjien vuorot muodostuvat aamupäivän kello 07-13, iltapäivän kello 13–19 ja 24/7-vuorosta. Nämä vuorot kestävät koko viikon, kuten taulukko 1 sitä kuvaa.

Kyseisiä päivystäjiä voi olla enemmänkin kuin taulukko kuvaa, joten interaktiivisen käyttöliittymän tarvitsee näyttää osa päivystäjätiedoista, eikä sen ole mahdollista näyttää kaikkea tietoa kerralla. Tämän vuoksi oli tärkeää tehdä komento, joka näyttää kaiken tiedon mutta tieto pysyy kompaktilla tasolla, jolloin puhelinta käyttävä henkilö sisäistää tiedon ylläpidon päivystäjistä ilman ongelmia.

NOC-komentoa suorittava henkilö saa kuvan 11 tapaisen vastauksen päivystäjistä. Käyttöliittymä näyttää keistä päivystäjistä puhutaan ensimmäisellä rivillä. Väliviivan jälkeen tulee tieto päivystäjistä ja mahdollisista lisätiedoista viikolle. Käyttöliittymä näyttää kolmea informaatiota, kuinka monennella sivulla olet kaikista päivystäjistä, viikko numeron ja päivämäärän. Alimmaisina olevilla painikkeilla voi mennä viikkoja eteenpäin, taaksepäin, hypätä nykyiselle viikolle tai piilottaa kaiken sisällön.



Kuva 11 NOC päivystäjät -komento Slack keskustelussa.

NOC päivystäjä -komennolle on mahdollista antaa viikkoparametri (kuva 12), joka määrittää minkä vastauksen Slack tulostaa keskusteluun. Tämä parametri tulee Boltin "command" objektin sisällä "text" arvossa. Esimerkiksi jos käyttäjä kirjoittaa `/dutyofficernoc 5`, hakee Slack-botti ensin kaiken päivystäjätiedon ja jäsentää tiedot YAML muodosta JSONiksi ja antaa tiedon viikon 5 päivystäjistä. Tätä viikkoparametria voidaan hyödyntää `dutyofficer` -komennon kanssa viikoilla 1–52. Jos viikkoa ei ole olemassa eli tieto puuttuu päivystäjä listasta, yrittää komento tarjota seuraavaa viikkoa

niin kauan kun päivystäjätietoja riittää funktiolla `nextHighestIndex` (kuva 12). Jos tämäkään ei ole mahdollista Slack-botti antaa loogisesti viimeisen päivystäjäviikon `index` tiedostossa. Lopuksi `respond` funktion vastaa lyhytaikaisella viestillä ja `block kit` palikalla. Viesti on näkymätön kaikille muille paitsi käyttäjälle, joka suorittaa komentoa.

```
app.command('/dutyofficernoc', async({ command, ack, respond }) => {
  const week = Number(command.text) || moment().week()
  ack()

  const response =
    YAML.parse(await dutyofficerFileUrl('data/officerDataNoc.yml'))

  const nextHighestIndex = (array, value, prop) => {
    for (let i = 0; i < array.length; i++){
      if (array[i]['date']['week'] >= value){
        return i;
      }
    }
    return array.length - 1;
  }

  const index = nextHighestIndex(response, week)

  respond({
    response_type: "ephemeral",
    blocks: messageBlockNoc(response, index)
  })
});
```

Kuva 12 NOC Slack komento

Funktiosta `messageBlockNoc`, joka sisältyy NOC komentoon on komennon käyttöliittymän liha. Käyttöliittymä koostuu noin sadasta rivistä Slack `block kit` JSONia ja pari konditionaalia painikkeiden rakennukseen, joka ottaa huomioon päivystäjä tiedon määrää ja tätä tietoa seuraten piilottaa tai / ja näyttää painikkeita tiedon. Kun huomioidaan päivystäjä tiedonmäärää, saadaan hyvä dynaaminen käyttöliittymä. Kuvassa 13 näkyy kolmen painikkeiden konditionaalinen osuus.

```

const messageBlockNoc = (officers, index) => {
  try {
    const maxlenght = officers.length - 1;

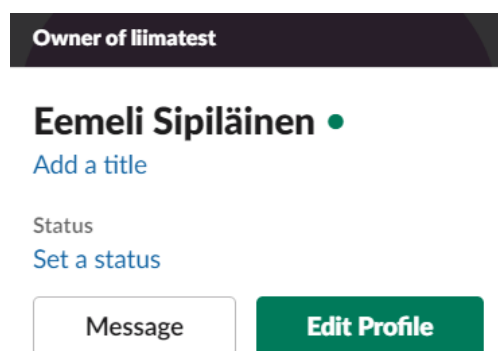
    let elements = []

    if(index <= maxlenght && index !== 0) {
      elements.push({
        "type": "button",
        "action_id": "dutyofficernoc:previousweek",
        "text": {
          "type": "plain_text",
          "text": `Previous week`
        },
        "value": `${index}`
      })
    }
    elements.push({
      "type": "button",
      "action_id": "dutyofficernoc:currentweek",
      "text": {
        "type": "plain_text",
        "text": `Current week`
      },
      "style": "primary",
      "value": `${moment().week()}`
    })
    if(index < maxlenght) {
      elements.push({
        "type": "button",
        "action_id": "dutyofficernoc:nextweek",
        "text": {
          "type": "plain_text",
          "text": `Next week`
        },
        "value": `${index}`
      })
    }
  }
  ...

```

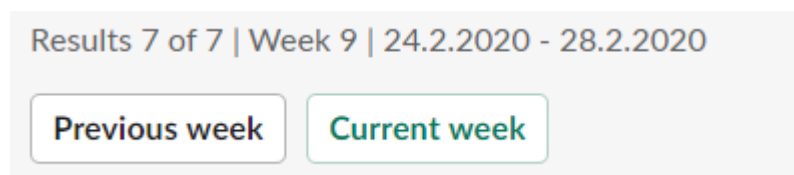
Kuva 13 Konditionaaliset NOC painikkeet, previous week, current week ja next week.

Päivystäjien nopea kontaktointi on yksi päätarkoituksista dutyofficernoc komennon suorituksessa. Päivystäjän nimestä klikkaamalla (@Eemeli Sipiläinen nimestä) kuvassa 14, avautuu ikkuna, jossa klikataan ”Viesti” -painikkeesta, jolloin voi lähettää nopeasti yksityisviestin päivystäjälle, eikä tarvitse kirjautua intranettiin hakemaan päivystäjätietoja.

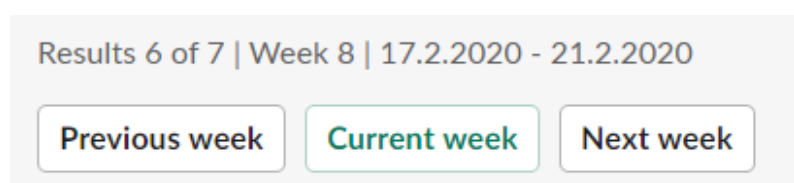


Kuva 14 Nimestä klikatun ikkunan sisältö.

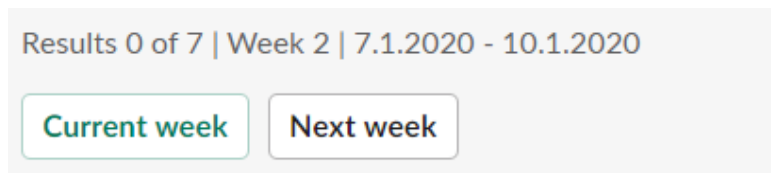
NOC päivystäjistä on kolme erilaista käyttöliittymäversioita. Käyttöliittymän eroavaisuudet tulevat esille tarkastelemalla kuvista 15-17 etenemiseen tarkoitettuja viikkopainikkeita.



Kuva 15 NOC päivystäjäkäyttöliittymäskenaario, jossa ollaan viimeisellä sivulla.



Kuva 16 NOC päivystäjäkäyttöliittymäskenaario, jossa ollaan alku- ja loppusivujen välissä.



Kuva 17 NOC päivystäjäkäyttöliittymäskenaario, jossa ollaan ensimmäisellä sivulla.

Näistä kaikissa painikkeissa on callback id ja action id, jonka avulla Bolt pystyy identifioimaan mistä klikkaus tulee ja tällä idllä vastaamaan tapahtumaan.

Kuva 18 kuvaa action id "dutyofficernoc:previousweek" päätepestettä. Tämä päätepeste odottaa klikkausta "previous week" -painikkeesta. Klikkauksen saatuaan päätepeste saa ack -funktion, respond -funktion ja payload -objektin. Ack -funktiolla hyväksytään tai hylätään tapahtuma funktion sisällä. Tämän tapahtuman aikaraja on noin pari sekuntia eli jos tässä ajassa ei suorita ack -funktiota niin tapahtuma hylätään. Payload sisältää monia Boltin muuttujia ja myös omani "value" merkkijonomuuttujan, joka määrittää indexiä päivystäjä tiedostossa ja tämän avulla valitsee edellisen päivystäjätiedon.

```
app.action('dutyofficernoc:previousweek', async({ ack, respond, payload }) => {
  const index = Number(payload.value) - 1
  ack()

  const response = YAML.parse(await dutyofficerFileUrl('data/officerDataNoc.yml'))

  respond({
    response_type: "ephemeral",
    blocks: messageBlockNoc(response, index)
  })
});
```

Kuva 18 NOC action previous week

Suurta eroa NOC päivystäjien tapahtuma painike koodien välillä ei ole. Erona on hylkäys (kuva 19) nykyisen viikon komentona, jos tieto on liian vanhaa.

```
app.action('dutyofficernoc:currentweek', async({ ack, respond }) => {
  const week = moment().week()

  const response =
    YAML.parse(await dutyofficerFileUrl('data/officerDataNoc.yml'))

  const officerIndex = response.map((e) =>
    { return e.date.week; }).indexOf(week)

  if(!response[officerIndex]) {
    ack('Error week not found')
    return
  }
  ack()

  respond({
    response_type: "ephemeral",
    blocks: messageBlockNoc(response, officerIndex )
  })
});
```

Kuva 19 NOC action current week

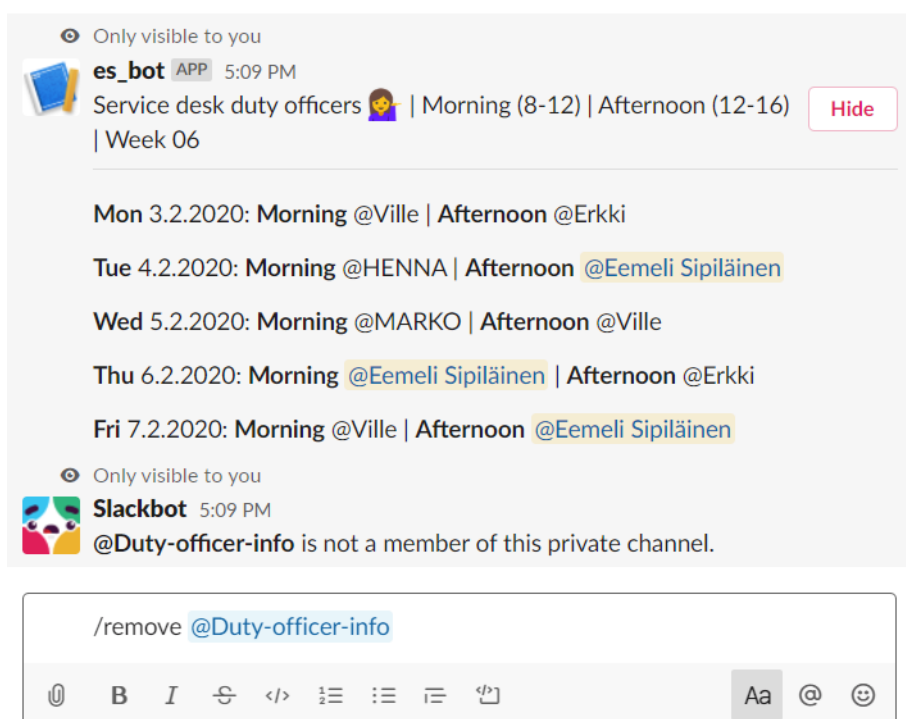
5.3.2 Asiakaspalvelun päivystäjäkäyttöliittymä

Toisena käyttöliittymäkokonaisuutena on asiakaspalvelun päivystäjät taulukko 2. Asiakaspalvelun päivystäjätiedot muodostuvat aamu 8–12 ja ilta-päivä 12–16 vuorosta. Tätä päivystystietoa on viikko kerrallaan ja jokaiselle päivälle on omat päivystäjänsä.

Asiakaspalvelun käyttöliittymäkokonaisuus on huomattavasti simppelempi kuin ylläpidon päivystäjien käyttöliittymä. Asiakaspalvelun käyttöliittymään pitää saada näkymään tietoa viikon joka päivä, joten kompakti

yhden sivun käyttöliittymäkokonaisuus on paras vaihtoehto päivystäjätiedoille (kuva 20).

Slack-botin toiminnan ja tietoturvan kannalta kaikki suoritettavat komennot vaativat minimimäärän OAuth rajoja toimiakseen. Eli toiminnaltaan selitettynä, jos käyttäjä kirjoittaa Slack-botti komennon kanavalla, on se sitten yksityinen tiimi/ryhmä, julkinen ryhmä/tiimi tai yksityinen viestikanaava, niin komennon toimivuus on taattua, koska botti lähettää lyhytaikaisen (ephemeral) viestin käyttäjälle. Tämä viesti on näkyvillä vain käyttäjälle, joka suoritti komennon kanavasta riippumatta. Kuva 20 kuvaa visuaalisesti tapahtumaa, kun käyttäjä suorittaa NOC päivystäjien komennon yksityisellä kanavalla.



Kuva 20 Asiakaspalvelun päivystäjät komento on suoritettu yksityisellä kanavalla ilman, että botti on lisätty sinne.

Asiakaspalvelun käyttöliittymä (kuva 21) syntyy silmukoiden section funktioita päivystäjätiedoilla. Päivystäjätiedot asetetaan header-taulukkoon jokainen section omana objektinaan.

```
const messageBlockSd = (officers) => {
  try {
    const section = (text) => {
      return {
        "type": "section",
        "text": {
          "type": "mrkdwn",
          "text": text
        }
      }
    }
  }
  const header = [
    {
      "type": "section",
      "text": {
        "type": "mrkdwn",
        "text": "Service desk duty officers :information_desk_person:
| Morning (8-12) | Afternoon (12-16) | Week 06"
      },
      "accessory": {
        "type": "button",
        "action_id": "dutyofficersd:hide",
        "text": {
          "type": "plain_text",
          "text": "Hide"
        }
      },
      "style": "danger"
    }
  ],
  {
    "type": "divider"
  },
  ]
  for (let i = 0; i < officers.length; i++) {
    const element = officers[i];
    header.push(
      section(`*${element.date.day}* ${element.date.date}:
*Morning* @${element.morning} |`

```

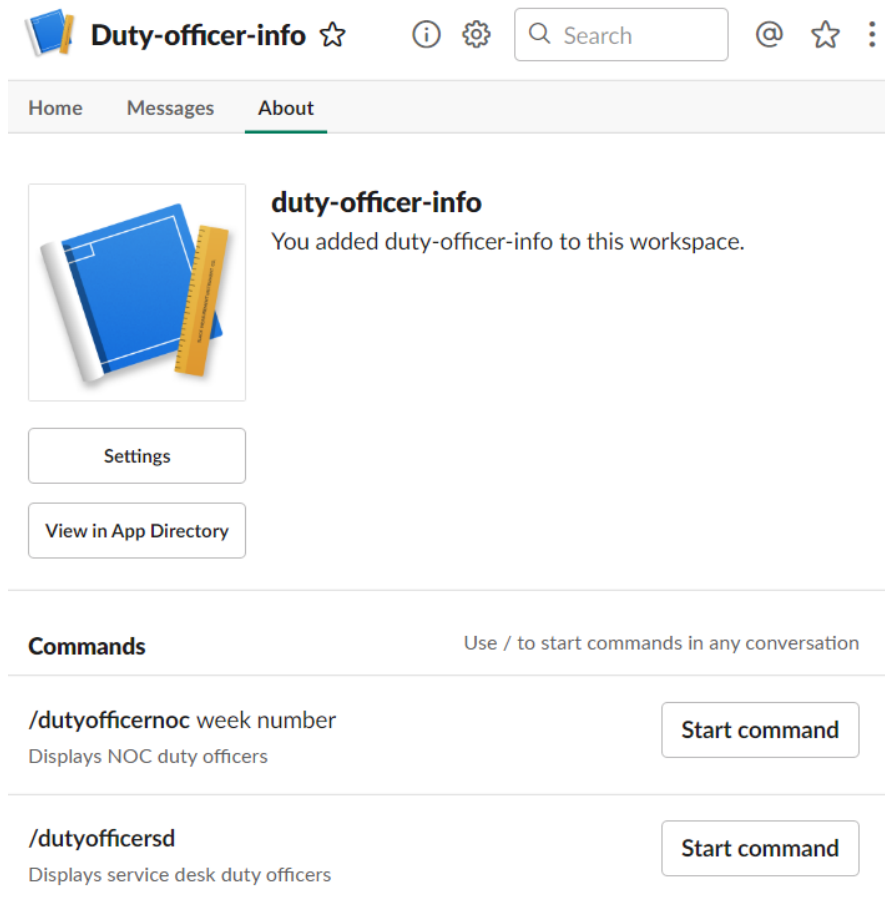
```
        *Afternoon* @${element.afternoon}`  
    )  
  )  
}  
return header  
} catch (error) { }  
}
```

Kuva 21 Slack sd komennon käyttöliittymä palikka.

Mietin vaihtoehtona monisivuisen käyttöliittymän tekemistä asiakaspalvelun päivystäjille ja samantapaisen painikkeen luomista kuin NOC päivystäjien käyttöliittymässä, jota voi selata seuraava viikko -painikkeella. Mutta päädyin siihen, että käyttäjältä veisi liian paljon aikaa selata sivuja, joissa on pelkästään päivittäisiä tietoja.

5.3.3 Muu käyttöliittymä

Botti ei tarjoa kommentia, joka näyttäisi kaikki bottiin rakennetut komennot, koska Slack on rakentanut tämän toiminnon itse "About" -välilehteen (kuva 22). Välilehti näyttää botin nimien, kuvauksen, komennot ja komentojen kuvauksen. Tästä välilehdestä käyttäjä saa hyvän kuvauksen ja ymmärryksen siitä mihin tarkoitukseen botti on rakennettu.



The screenshot shows the Slack interface for a workspace named "duty-officer-info". At the top, there is a search bar and navigation icons. Below the workspace name, a message states "You added duty-officer-info to this workspace." There are two buttons: "Settings" and "View in App Directory". A "Commands" section follows, with the instruction "Use / to start commands in any conversation". Two commands are listed:

| Command | Description | Action |
|--|-------------------------------------|---------------|
| <code>/dutyofficernoc</code> week number | Displays NOC duty officers | Start command |
| <code>/dutyofficersd</code> | Displays service desk duty officers | Start command |

Kuva 22 Slack-botti kuvaus -välilehti kuvaa botin käyttötarkoitusta ja komentoja.

5.4 Botin käytettävyydestä

Botin käytettävyydestä varten toteutin sitä koskevan englanninkielisen kyselyn nimeltään "Päivystäjän Slack-botti – kysely". Kyselyssä pyysin Ambientian testaa- jia arvioimaan botin toiminnallisuutta, käyttökelpoisuutta ja visuaalisuutta. Lisäksi pyysin ilmoittamaan mahdollisista ongelmista botin käytössä ja toivoin parannusehdotuksia ja muita botin käyttöön liittyviä kommentteja. Ilmoitin, että kyselyn vastauksia käytetään osana opinnäytetyötäni. Kysymyksiä oli yhteensä 11 kappaletta. Vastaajia oli seitsemän.

5.4.1 Kyselyn tulokset

Kyselyn kysymykset ja vastaukset sekä palaute testaajilta ovat alla.

Kuinka hyödyllinen NOC-päivystäjätietoja esittävä käyttöliittymä mielestäsi oli?

- keskiarvo 4,14 ja seitsemän vastaajaa

7
vastausta



Kuinka hyödyllinen Service Desk -päivystäjätietoja esittävä käyttöliittymä mielestäsi oli?

- keskiarvo 4,00 ja seitsemän vastaajaa

7
vastausta



Saitko Service Desk -päivystäjistä sellaiset vastaukset kuin odotit?

- seitsemän vastaajaa ja kaikki saivat odottamansa vastauksen

| | |
|-------|---|
| ● Yes | 7 |
| ● No | 0 |



Mitä vastauksia odotit näkeväsi Service Desk -päivystäjistä?

- odotin näkeväni, kuka on päivystämässä tällä viikolla
- odotin näkeväni listan kuluvan viikon SD-päivystäjistä

2
vastausta

Uusimmat vastaukset
"To see who are on call this week"
"I expected to see SD duty officers listed for the current week"

Mitä vastauksia odotit näkeväsi NOC-päivystäjistä?

- odotin näkeväni, kuka on päivystämässä tällä viikolla
- odotin näkeväni listan kuluvaan viikon NOC-päivystäjistä

2
vastausta

Uusimmat vastaukset
"To see who are on call this week."
"I expected to see NOC duty officers listed for the current week"

Mitä mieltä olet Service Desk -päivystäjätietokäyttöliittymästä? Piditkö sen ulkonäöstä?

- keskiarvo 3,57 ja seitsemän vastaajaa

7
vastausta



Mitä mieltä olet NOC-päivystäjätietokäyttöliittymästä? Piditkö sen ulkonäöstä?

- keskiarvo 4,14 ja seitsemän vastaajaa

7
vastausta



Oliko sinulla ongelmia botin käytössä?

- kuudella ei ollut ongelmia ja yhdellä oli vähän ongelmia

| | |
|----------|---|
| ● No | 6 |
| ● Little | 1 |
| ● A lot | 0 |



Jos sinulla oli ongelmia, niin kerro millaisia.

- /dutyofficer on hieman pitkä ja kömpelö sana, ehdottaisin /sdoncall ja /noconcall tai komennon parametrisointia /"oncall sd" ja /"oncall noc"
- näyttää myös siltä, että Slack @ ei toimi tällä hetkellä, jos nimeen kirjoitetaan ääö
- lisäksi on hieman epäselvää mitä Notice: means in SF on call list

Vastaajien parannusehdotukset tai muut kommentit.

- En koskaan muista viikon numeroita, mutta kun olen tajunnut, että botti näyttää oletuksena nykyisen viikon, niin se on täydellinen.
- Tietämättä Slack-botti-käyttöliittymän räätälöinnin rajoituksia; SD-tuloksia on vaikea lukea: tietoa on paljon. Pystysuuntainen yhdenmu-kaistaminen informaatio suunnittelussa auttaisi, ovatko taulukot vaihto-eto. Korostaisin myös nykyisen päivän riviä. Sama pätee NOC-tuloksiin, mutta vähäisemmässä määrin, johtuen tiedon määrästä. SD- ja NOC-tulosten kuvausrivit eroavat toisistaan; näiden yhtenäisyys olisi hyvä, ja alustaisin usean rivin tiedot sen sijaan, että käyttäisin | sym-bolia.
- Ehkä botti voisi tarjota linkkejä joihinkin ohjeisiin siitä, millaisia asioita päivystysalan ihmisten tulisi hoitaa. Hyvä botti!
- Viikon muistiinpanot ovat "nolla" / dutyofficernoc-näkymässä. Voidaan piilottaa, jos muistiinpanoja ei ole?

6 JOHTOPÄÄTÖKSET

Opinnäytetyön tarkoituksena oli luoda päivystäjä - Slack-botti Ambientia Group Oy:n verkkotoimintojen keskuksen ylläpidon päivystäjien ja asiakas-palvelun päivystäjien päivystäjätietojen esille hakemiseen ja esittämiseen. Slack-botin tarkoitus oli hakea päivystäjätiedot esiin ja kertoa kuka päivys-täjä kulloinkin on päivystysvuorossa, samalla sovellus mahdollistaisi viestin lähettämisen päivystäjälle.

Slack-botin käytännön työn osuudessa onnistuttiin hyvin ja botti tuli onnis-tuneesti koodattua järjestelmään käyttäen siihen tarkoitettuja teknologi-
oita, työkaluja ja menetelmiä, joita on selostettu tässä opinnäytetyössä.

Kehittämiskyselyn kautta saadut arviot botista olivat myönteisiä. Botin toimi niin kuin oli tarkoitettu ja toteutti tehtävänsä. Yleisarvosana oli 4,14 asteikon ollessa 1–5.

Samalla opinnäytetyön käytännön osuus osoitti, että Slack-botti on hyödyllinen työkalu erilaisten tehtävien automatisoinnissa, tässä tapauksessa päivystäjätietojen haun automatisoinnissa. Lisäksi osoitettiin, että Slack -alusta toimii hyvin botin rakentamisessa ja kehittämisessä.

Jatkossa bottia voisi rakentaa dynaamisempaan suuntaan. Esimerkiksi uuden päivystäjätiedon lisääminen bottia hyödyntämällä voisi olla hyödyllinen lisätoiminto.

Teoriaosuudessa tuli esille, että erityisesti vuorovaikutteisten viesti- ja bottijärjestelmien suosio kasvaa jatkuvasti ja myös ohjelmistobottien käytön odotetaan lisääntyvän ja johtavan parannuksiin kehittäjien ja tiimien tuottavuudessa ja tehokkuudessa.

LÄHTEET

Ambientia. (n.d.). Haettu 23.1.2020 osoitteesta <https://www.ambientia.fi/>

Anderson, K.E. (2016). Getting acquainted with social networks and apps: picking up the Slack in communication and collaboration. 11.7.2016 Haettu osoitteesta <https://rucore.libraries.rutgers.edu/rutgers-lib/51298/PDF/1/>

Malcolm, F., Pring, B. & Roehrig, P. (2017). What to Do When Machines Do Everything: How to Get Ahead in a World of AI, Algorithms, Bots, and Big Data. John Wiley & Sons, Incorporated. <https://ebookcentral-proquest-com.ezproxy.hamk.fi/lib/hamk-ebooks/reader.action?docID=4790360&ppg=7>

Elliott, A. (2018) The Culture of AI: Everyday Life and the Digital Revolution, Routledge. ProQuest Ebook Central. Haettu 16.2.2020 osoitteesta <https://ebookcentral-proquest-com.ezproxy.hamk.fi/lib/hamk-ebooks/detail.action?docID=5613372>

Eronen, R. (n.d.). Kielikello, Kielenhuollon tiedotuslehti. Haettu 7.2.2020 osoitteesta <https://www.kielikello.fi/-/hyvat-ja-pahat-botit>

Gentsch, P. (2018) AI in Marketing, Sales and Service: How Marketers Without a Data Science Degree Can Use AI, Big Data and Bots. Palgrave Macmillan US, 2018. ProQuest Ebook Central. Haettu 21.2.2020 osoitteesta <http://ebookcentral.proquest.com/lib/hamk-ebooks/detail.action?docID=5566829>

Phillips. C. (2018). The 3 Types of Chatbots & How to Determine the Right One for Your Needs. Blogijulkaisu 30.4.2018. Haettu 24.1.2020 osoitteesta

<https://chatbotsmagazine.com/the-3-types-of-chatbots-how-to-determine-the-right-one-for-your-needs-a4df8c69ec4c>

Slack. (n.d.). Bolt: a foundational framework for Slack apps. Haettu 26.2.2020 osoitteesta <https://api.slack.com/start/building#frameworks>

Slack. (n.d.). Bot users. Haettu 24.1.2020 osoitteesta <https://api.slack.com/bot-users>

Slack. (n.d.). Build – The art of creating a Slack app your team will love Haettu 31.1.2020 osoitteesta <https://slack.dev/guides/Build.pdf>

Slack. (n.d.). Building a Slack app. Haettu 31.1.2020 osoitteesta <https://api.slack.com/start/building>

Slack. (n.d.). What is Slack. Haettu 23.1.2020 osoitteesta <https://slack.com/intl/en-fi/help/articles/115004071768-What-is-Slack->

Spasovski, M. (2013). OAuth 2.0 -identiteetti- ja pääsyhallintamallit, Packt Publishing, Limited. ProQuest Ebook Central. Haettu 16.2.2020 osoitteesta <http://ebookcentral.proquest.com/lib/hamk-ebooks/detail.action?docID=1572907>

Storey. M-A & Zagalsky. A. (2016). Disrupting Developer Productivity One Bot at a Time. 24.11.2016. Haettu osoitteesta <http://chisel.cs.uvic.ca/pubs/storey-FSE-VaR2016.pdf>

Teixeira P. (2013). Instant Node.js Starter. Packt Publishing, Limited. Haettu 16.2.2020 osoitteesta <http://ebookcentral.proquest.com/lib/hamk-ebooks/detail.action?docID=1214993>

Yaml. (n.d.) YAML Ain't Markup Language (YAML™) Version 1.2. Haettu
26.2.2020 osoitteesta <https://yaml.org/spec/1.2/spec.html>

Liite 1: Duty officer Slack Bot- survey

Duty officer Slack Bot - survey

Hi there. Please respond to this Slack bot duty officer survey. I would like your evaluation of the functionality, usability, visuality and possible bot problems of the Bot. I would also appreciate any suggestions for improvement or any other bot related comments. Word is free. Thank you in advance for your time, reviews, and comments. Yours sincerely, Eemeli Sipiläinen. HAMK tietojenkäsittelijä.

These stats will be used for my thesis, thank you.

* Pakollinen

* Lomake tallentaa nimesi. Kirjoita nimesi.

1. How useful do you think the interface was for network operations
centre duty officer command /dutyofficernoc *



2. How useful do you think the interface was for service desk duty officer
command/dutyofficersd *



3. Did you get the answers you were expecting? for service desk duty officer command/dutyofficersd *

Yes

No

4. What answers did you expect? Service desk

5. Did you get the answers you were expecting? for network operations centre duty officer command /dutyofficernoc *

Yes

No

6. What answers did you expect? Network operations centre

7. Your appreciation of the visual appearance of the service desk user interface, did you like its appearance? *



8. Your appreciation of the visual appearance of the network operations

centre user interface, did you like its appearance? *



9. Were there problems with bot usage? *

- No
- Little
- A lot

10. Could you please tell me problems if any?

11. Suggestions for improvement or any other bot related comments!

Tämä ei ole Microsoftin luomaa tai suosittelemaa sisältöä. Lähettämäsi tiedot lähetetään lomakkeen omistajalle.

 Microsoft Forms

11/22/2020