

Tekninen dokumentaatio

ChatBot

Joni Lassila, Emil Ålgars

Metropolia Ammattikorkeakoulu
Insinööri (AMK)
Koulutusohjelma
Insinöörityö
Päivämäärä

Sisällys

Contents

| | | |
|---|---|---|
| 1 | Johdanto | 1 |
| 2 | Järjestelmäarkkitehtuuri | 1 |
| | 2.1 Käyttöliittymäkerros | 1 |
| | 2.1.1 Käyttöliittymäkerroksen tehtävät | 1 |
| | 2.1.2 Javafx-kirjaston käyttö | 2 |
| | 2.2 Sovelluslogiikkakerros | 2 |
| | 2.3 Tietokantakerros | 2 |
| 3 | Tietokanta | 2 |
| 4 | Testaus | 3 |
| | 4.1 Manuaalinen testaus | 3 |
| | 4.2 Automaattinen testaus | 3 |
| 5 | Jatkokehitysideoita | 4 |

Käyttöohje ChatBot

1 Johdanto

Tämä tekninen dokumentaatio kuvaa ChatBot sovelluksen rakenteen, arkkitehtuurin ja testauksen.

2 Järjestelmäarkkitehtuuri

ChatBot sovellus on toteutettu Java ohjelmointikielellä ja käyttää MySQL tietokantaa kysymysten ja vastausten tallentamiseen. Sovellus noudattaa kerrosarkkitehtuuria, jossa eri toiminnallisuudet on jaettu kolmeen pääkerrokseen:

1. Käyttöliittymäkerros
2. Sovelluslogiikkakerros
3. Tietokantakerros

2.1 Käyttöliittymäkerros

2.1.1 Käyttöliittymäkerroksen tehtävät

Käyttöliittymän tehtävä on näyttää käyttäjälle ikkuna, joka helpottaa ja parantaa käyttäjän kokemusta chatbotin kanssa. Käyttöliittymäkerros vastaanottaa käyttäjän syötteitä ja välittää niitä sovelluksen muille kerroksille. Sen tehtävänä on myös esittää käyttäjälle sovelluksen käyttöliittymä, joka voi sisältää erilaisia komponentteja, kuten nappeja, tekstikenttiä, ikkunoita ja muita käyttöliittymän elementtejä.

2.1.2 Javafx-kirjaston käyttö

Javafx-kirjaston käyttö tarjosi erinomaisen työkalun sovelluksen käyttöliittymäkerroksen toteuttamiseen. Kirjasto tarjosi monipuolisia komponentteja ja työkaluja, jotka helpottivat käyttöliittymän suunnittelua ja toteuttamista. Kirjaston käyttö mahdollisti myös tyylien ja ulkoasun hallinnan, joka oli tärkeä osa chatbotin käyttäjäkokemuksen optimointia.

2.2 Sovelluslogiikkakerros

Sovelluslogiikkakerros vastaa sovelluksen keskeisistä toiminnoista, kuten kysymysten ja vastausten käsittelystä, salasanojen salauksesta ja tietokantaoperaatiosta.

2.3 Tietokantakerros

Tietokantakerros vastaa tietokantayhteyksistä ja tietokantaoperaatioista, kuten kysymysten ja vastausten lisäämisestä, muokkaamisesta ja poistamisesta sekä admin käyttäjien hallinnasta.

3 Tietokanta

Sovelluksen tietokanta on toteutettu MySQL tietokantajärjestelmällä. Tietokanta koostuu kahdesta taulusta, joka sisältää kysymykset ja vastaukset, sekä admin tiedot.

Kysymykset ja vastaukset taulukko (QA) on seuraavanlainen:

- Id: Yksilöllinen tunniste kysymykselle.
- Question: Kysymyksen teksti.
- Answer: Vastauksen teksti.

Admin taulukko on seuraavanlainen:

- Username: Admin käyttäjän käyttäjätunnus.
- Hash: Salatun salasanan hajautus.
- Salt: Käytetty ”suola” salasanan salauksessa.

4 Testaus

Sovelluksen testaus koostuu manuaalisista sekä automaattisista testeistä. Tämä kappale kuvaa näitä testausmenetelmiä tarkemmin.

4.1 Manuaalinen testaus

Manuaalinen testaus kattaa sovelluksen käyttöliittymän ja toiminnallisuuden testaamisen. Manuaalisilla testeillä tarkistettiin, että sovelluksen eri osat toimivat oikein ja täytävät vaatimusmäärittelyt. Manuaalisessa testauksessa keskityttiin erityisesti seuraaviin asioihin:

- Käyttöliittymän ulkoasu ja toiminta eri ikkunakoilla.
- Kysymysten ja vastausten lisääminen, muokkaaminen ja poistaminen.
- Virheellisten syötteiden käsittely ja virheilmoitukset

4.2 Automaattinen testaus

Suoritamme automaattisen testauksen jenkins-ohjelmistoa käyttäen. Jokaiselle ohjelmistolle on oma jenkins työ, joka ajaa testit gitlabin main haarasta. Työ alustaa työtilan tyhjentämällä sen. Sitten se hyödyntää jenkinsin "Environment Injector" laajennusta asettaakseen työn ympäristömuuttujat, joita maven tarvitsee rakentaakseen sovelluksen. Tämän jälkeen maven-surefire lisäosa suorittaa yksikkötestauksen ja Jacoco luo koodin kattavuus raportin. Tämän jälkeen testien, ja koodin kattavuus raportit tulevat näkyviin jenkinsiin.

Yksikkötestit ovat kirjoitettu käyttäen Junit5 kirjastoa, ja hyödyntäen muiden testaus kirjastojen toimintoja (esimerkiksi: mockito, mockMVC). Yksikkötesteitä on suurimmille osille logiikkakerroksen toiminnoista.

5 Jatkokehitysideoita

Nämä ideat auttavat laajentamaan chatbotin toimintoja ja parantamaan sen käyttäjäkokemusta. On tärkeää arvioida, mitkä ideat sopivat parhaiten chatbotin tarkoitukseen ja käyttäjäkuntaan, jotta chatbotin kehitys onnistuu parhaiten.

- Monikielinen tuki
- "Text-to-Speech"
- Datan keräys, ja analysointi