PseudoCode Chatbot: SILLY

(Straightforward Interactive Lingustic Language Yapper)

```
Funktion issue_match(keywords):
    Falls keywords leer:
        keywords = Nutzer-Eingabe erhalten
    issue = keyword_search(keywords)

    Falls issue ist "update":
        Bot-Nachricht: "Haben Sie Probleme mit dem Update? (ja/nein)"
        Falls clarification(keywords, "update") == true:
            case_update()

    Falls issue ist "return":
        Bot-Nachricht: "Möchten Sie Ihr Produkt zurückgeben? (ja/nein)"
        Falls clarification(keywords, "return") == true:
            case_return()

    Falls issue ist "agent":
        Bot-Nachricht: "Möchten Sie mit einem Support-Agenten sprechen? (ja/nein)"
        Falls clarification(keywords, "agent") == true:
            Bot-Nachricht: "Ein Ticket wird erstellt."
            case_agent()

    Falls issue ist "EXIT":
        Bot-Nachricht: "Es tut mir leid, dass ich nicht helfen konnte."
        Log: "Gespräch beendet, Ticket wird verworfen."

    Sonst:
        Bot in "silent mode" starten
        Bot-Nachricht: "Ich habe das nicht verstanden. Bitte erklären Sie Ihr Proble
        issue_match()
```

```
Funktion issue_match(keywords):
    Falls keywords leer:
        keywords = Nutzer-Eingabe erhalten
    issue = keyword_search(keywords)

    Falls issue ist "update":
        Bot-Nachricht: "Haben Sie Probleme mit dem Update? (ja/nein)"
        Falls clarification(keywords, "update") == true:
            case_update()

    Falls issue ist "return":
        Bot-Nachricht: "Möchten Sie Ihr Produkt zurückgeben? (ja/nein)"
        Falls clarification(keywords, "return") == true:
            case_return()

    Falls issue ist "agent":
        Bot-Nachricht: "Möchten Sie mit einem Support-Agenten sprechen? (ja/nein)"
        Falls clarification(keywords, "agent") == true:
            Bot-Nachricht: "Ein Ticket wird erstellt."
            case_agent()

    Falls issue ist "EXIT":
        Bot-Nachricht: "Es tut mir leid, dass ich nicht helfen konnte."
        Log: "Gespräch beendet, Ticket wird verworfen."

    Sonst:
        Bot in "silent mode" starten
        Bot-Nachricht: "Ich habe das nicht verstanden. Bitte erklären Sie Ihr Proble
        issue_match()
```

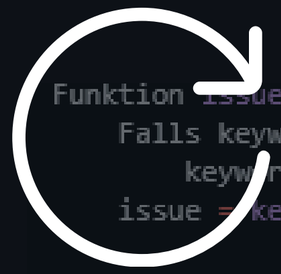
```
Funktion issue_match(keywords):
    Falls keywords leer:
        keywords = Nutzer-Eingabe erhalten
    issue = keyword_search(keywords)

    Falls issue ist "update":
        Bot-Nachricht: "Haben Sie Probleme mit dem Update? (ja/nein)"
        Falls clarification(keywords, "update") == true:
            case_update()

    Falls issue ist
        Bot-Nachrich
        Falls clarif
            case_ret

    Falls issue ist
        Bot-Nachrich
        Falls clarif
            Bot-Nach
            case_age

    Falls issue ist "EXIT":
        Bot-Nachricht: "Es tut mir leid, dass ich nicht helfen konnte."
        Log: "Gespräch beendet, Ticket wird verworfen."

    Sonst:
        Bot in "silent mode" starten
        Bot-Nachricht: "Ich habe das nicht verstanden. Bitte erklären Sie Ihr Proble
        issue_match()
```

```
Funktion keyword_search(suchbegriff):
Falls suchbegriff nicht "EXIT" ist:
    Konvertiere suchbegriff zu Kleinbuchstaben und erstelle eine Wortliste
    gefundene_kategorie = keine
    Für jedes Thema, Synonyme in der Themenliste:
        Falls ein Wort aus suchbegriff in den Synonymen enthalten ist:
            gefundene_kategorie = Thema
            Gib gefundene_kategorie zurück, Schleife wird beendet
Sonst:
    Suche abbrechen
```

```
Funktion issue_match(keywords):
    Falls keywords leer:
        keywords = Nutzer-Eingabe erhalten
    issue = keyword_search(keywords)

    Falls issue ist "update":
        Bot-Nachricht: "Haben Sie Probleme mit dem Update? (ja/nein)"
        Falls clarification(keywords, "update") == true:
            case_update()

    Falls issue ist "return":
        Bot-Nachricht: "Möchten Sie Ihr Produkt zurückgeben? (ja/nein)"
        Falls clarification(keywords, "return") == true:
            case_return()

    Falls issue ist "agent":
        Bot-Nachricht: "Möchten Sie mit einem Support-Agenten sprechen? (ja/nein)"
        Falls clarification(keywords, "agent") == true:
            Bot-Nachricht: "Ein Ticket wird erstellt."
            case_agent()

    Falls issue ist "EXIT":
        Bot-Nachricht: "Es tut mir leid, dass ich nicht helfen konnte."
        Log: "Gespräch beendet, Ticket wird verworfen."

    Sonst:
        Bot in "silent mode" starten
        Bot-Nachricht: "Ich habe das nicht verstanden. Bitte erklären Sie Ihr Proble
        issue_match()
```

```
Funktion issue_match(keywords):
    Falls keywords leer:
        keywords = Nutzer-Eingabe erhalten
    issue = keyword_search(keywords)

    Falls issue ist "update":
        Bot-Nachricht: "Haben Sie Probleme mit dem Update? (ja/nein)"
        Falls clarification(keywords, "update") == true:
            case_update()

    Falls issue ist "return":
        Bot-N
        Falls
            c

    Falls iss
        Bot-N
        Falls
            B
            case_agent()

    Falls issue ist "EXIT":
        Bot-Nachricht: "Es tut mir leid, dass ich nicht helfen konnte."
        Log: "Gespräch beendet, Ticket wird verworfen."

    Sonst:
        Bot in "silent mode" starten
        Bot-Nachricht: "Ich habe das nicht verstanden. Bitte erklären Sie Ihr Proble
        issue_match()
```

```
Funktion clarification(keywords, Thema):
    Falls keyword_search(Nutzer-Eingabe) nicht "yes":
        Lösche das Thema aus der Themenliste
        Bot-Nachricht: "Danke für die Klarstellung. Ich werde die Eingabe erneut verarbeiten."
        issue_match(keywords)
    Sonst:
        Rückgabe true
```



```
Funktion issue_match(keywords):
    Falls keywords leer:
        keywords = Nutzer-Eingabe erhalten
    issue = keyword_search(keywords)

    Falls issue ist "update":
        Bot-Nachricht: "Haben Sie Probleme mit dem Update? (ja/nein)"
        Falls clarification(keywords, "update") == true:
            case_update()

    Falls issue ist "return":
        Bot-N
        Falls
            c

    Falls iss
        Bot-N
        Falls
            B
            case_agent()

    Falls issue ist "EXIT":
        Bot-Nachricht: "Es tut mir leid, dass ich nicht helfen konnte."
        Log: "Gespräch beendet, Ticket wird verworfen."

    Sonst:
        Bot in "silent mode" starten
        Bot-Nachricht: "Ich habe das nicht verstanden. Bitte erklären Sie Ihr Proble
        issue_match()
```

```
Funktion clarification(keywords, Thema):
    Falls keyword_search(Nutzer-Eingabe) nicht "yes":
        Lösche das Thema aus der Themenliste
        Bot-Nachricht: "Danke für die Klarstellung. Ich werde die Eingabe erneut verarbeiten."
        issue_match(keywords)
    Sonst:
        Rückgabe true
```

geht wieder in die Funktion,
macht keyword_search Thema weniger, sucht nach nächsten match

z.B. bei Eingabe von update return, update aus dictionary gelöscht, also return

```
Funktion issue_match(keywords):
    Falls keywords leer:
        keywords = Nutzer-Eingabe erhalten
    issue = keyword_search(keywords)

    Falls issue ist "update":
        Bot-Nachricht: "Haben Sie Probleme mit dem Update? (ja/nein)"
        Falls clarification(keywords, "update") == true:
            case_update()

    Falls issue ist "retur
        Bot-Nachricht: "Mö
        Falls clarificatic
            case_return()

    Falls issue ist "agent
        Bot-Nachricht: "Mö
        Falls clarificatic
            Bot-Nachricht:
            case_agent()

    Falls issue ist "EXIT":
        Bot-Nachricht: "Es tut mir leid, dass ich nicht helfen konnte."
        Log: "Gespräch beendet, Ticket wird verworfen."

    Sonst:
        Bot in "silent mode" starten
        Bot-Nachricht: "Ich habe das nicht verstanden. Bitte erklären Sie Ihr Proble
        issue_match()
```

```
Funktion case_update():
    Senden der Update-Anleitung
    Frage nach Zufriedenheit
    Falls keyword_search(Nutzer-Eingabe) nicht "yes":
        Bot-Nachricht: "Ein Ticket wird erstellt."
        case_agent("update")
    Sonst:
        Bot-Nachricht: "Danke für die Kontaktaufnahme."
        Log: "Problem gelöst, kein Ticket notwendig."
```

d.h. Eingabe des Nutzers ist weder yes noch Synonym von yes

```
Funktion issue_match(keywords):
    Falls keywords leer:
        keywords = Nutzer-Eingabe erhalten
    issue = keyword_search

    Falls issue ist "updat
        Bot-Nachricht: "Ha
        Falls clarificatic
            case_update()

    Falls issue ist "retur
        Bot-Nachricht: "Mö
        Falls clarificatic
            case_return()

    Falls issue ist "agent":
        Bot-Nachricht: "Möchten Sie mit einem Support-Agenten sprechen? (ja/nein)"
```

```
Funktion case_update():
    Senden der Update-Anleitung
    Frage nach Zufriedenheit
    Falls keyword_search(Nutzer-Eingabe) nicht "yes":
        Bot-Nachricht: "Ein Ticket wird erstellt."
        case_agent("update")
    Sonst:
        Bot-Nachricht: "Danke für die Kontaktaufnahme."
        Log: "Problem gelöst, kein Ticket notwendig."
```

```
Funktion case_agent(topic):
    Falls kein Thema gegeben:
        Setze Thema auf "misc" //für miscommunication
    email = leer
    Solange die Schleife nicht beendet wird:
        Bot-Nachricht: "Geben Sie Ihre E-Mail-Adresse ein oder lassen Sie das Feld leer, um zu überspringen."
        email = Benutzereingabe ohne Leerzeichen an Anfang und Ende, die dem Log hinzugefügt wird
        Wenn email = leer oder nach Funktion validate_email gültig:
            Schleife beenden
        Sonst:
            Bot-Nachricht: "Falsches Mail Format. Bitte geben Sie eine gültige Mail an"
```

Fall

Sons

.

```
Funktion case_agent(topic):
    Falls kein Thema gegeben:
        Setze Thema auf "misc" //für miscommunication
    email = leer
    Solange die Schleife nicht beendet wird:
        Bot-Nachricht: "Geben Sie Ihre E-Mail-Adresse ein oder lassen Sie das Feld leer, um zu überspringen."
        email = Benutzereingabe ohne Leerzeichen an Anfang und Ende, die dem Log hinzugefügt wird
        Wenn email = leer oder nach Funktion validate_email gültig:
            Schleife beenden
        Sonst:
            Bot-Nachricht: "Falsches Mail Format. Bitte geben Sie eine gültige Mail an"

    Solange die Schleife nicht beendet wird:
        [...] //dasselbe mit Telefonnummer

    Wenn weder email noch Telefonnummer angegeben:
        Bot-Nachricht: "Du musst mindestens einen Kontaktweg angeben..."
        rufe Funktion case_agent() auf

    preference = leer
    Wenn sowohl email als auch Telefonnummer angegeben:
        Solange die Schleife nicht beendet wird:
            Bot-Nachricht: "Welchen Kontaktweg bevorzugst du?"
            preference_input = Benutzereingabe ohne Leerzeichen an Anfang und Ende, die dem Log hinzugefügt wird
            matched_preference = keyword_search(preference_input)

            Wenn matched_preference email oder Telefonnummer ist:
                preference = matched_preference
                Schleife beenden
            Sonst:
                Bot-Nachricht: "Bitte email oder Telefonnummer angeben"

    Rufe Funktion create_ticket(mit ticket_id, Thema, email, Telefonnummer, Bevorzugtem Kontaktaweg)
    Nachricht: "Ticket wurde generiert, wir melden uns"
```