AASD 24Z projekt

- prowadzący: Tomasz Śliwiński
- zespół: MegaAgentyAI
- skład zespołu:
  - Marcel Tracz
  - Maksim Makaranka
  - Jan Retkowski
  - Miłosz Łopatto
- repozytorium: https://github.com/milosz-l/AASD

Spis treści

1. Dokumentacja wstępna: raport A - identyfikacja problemu

1.1. Analiza burzy mózgów

Podczas burzy mózgów skoncentrowano się na tworzeniu koncepcji systemów, które mogłyby wspierać rozwój "mądrego miasta". W trakcie dyskusji zaproponowano różne rozwiązania, mające na celu ułatwienie życia mieszkańcom i odwiedzającym miasto poprzez optymalizację i integrację dostępnych usług.

Jednym z rozważanych pomysłów był system do identyfikacji i zarządzania problemami z infrastrukturą miejską. System ten miał umożliwiać mieszkańcom zgłaszanie problemów, takich jak awarie czy uszkodzenia, przy jednoczesnym sprawdzeniu ich statusu i priorytetu. Ostatecznie pomysł ten został odrzucony, ponieważ podobne

aplikacje już funkcjonują w wielu miastach, efektywnie obsługując tego rodzaju zgłoszenia, np. w przypadku Warszawy istnieje serwis dostępny pod linkiem.

Innym pomysłem było stworzenie informatora o wydarzeniach i ciekawych miejscach dla turystów i mieszkańców. Projekt ten miał na celu dostarczanie spersonalizowanych informacji o lokalnych wydarzeniach i atrakcjach, bazując na zainteresowaniach użytkowników i ich bieżącej lokalizacji. Zrezygnowano jednak z tego rozwiązania z uwagi na dużą liczbę dostępnych już narzędzi oferujących podobne usługi.

1.2. Identyfikacja i opis problemu

Współczesne społeczeństwo stoi przed wyzwaniem szybkiego rozprzestrzeniania się chorób zakaźnych, co często wyprzedza możliwości ich szybkiego wykrycia i kontrolowania. Brakuje zintegrowanego systemu monitorującego objawy chorób w czasie rzeczywistym, co ogranicza zdolność do szybkiej identyfikacji potencjalnych ognisk epidemicznych.

Dodatkowym problemem jest brak narzędzi umożliwiających ludziom łatwe zgłaszanie objawów oraz otrzymywanie wstępnej oceny ich znaczenia. Bez takiego systemu trudno jest gromadzić i analizować dane na temat zdrowia publicznego w sposób skuteczny. Potrzebne jest rozwiązanie, które nie tylko umożliwi zbieranie i analizowanie danych o objawach z różnych lokalizacji, ale także wspierać będzie służby zdrowia w identyfikacji zagrożeń epidemicznych i reagowaniu na nie w odpowiednim czasie.

1.2.1. Interesariusze:

- Służby zdrowia (np. sanepid, instytuty epidemiologiczne)
- Władze lokalne i krajowe (np. ministerstwa zdrowia)
- Badacze i naukowcy (specjaliści zajmujący się epidemiologią)
- Pracownicy służb ratunkowych (np. zespoły medyczne, ratownicy)
- Pacjenci (osoby dotknięte zakażeniem)
- Lekarze
- Dostawcy systemów danych medycznych (bazy danych, aplikacje zdrowotne)
- Specjaliści ds. epidemiologii (analitycy zajmujący się danymi o chorobach)

1.2.2. Zastosowania:

- Dokonywanie wspępnej diagnozy na podstawie objawów
- Pomaganie specjalistom w diagnozowaniu pacjentów
- Polecanie użytkownikom do jakiej specjalizacji lekarza się udać
- Lokalizacja ognisk zakażeń chorób zakaźnych

1.3. Propozycja i sprecyzowanie rozwiązania

W obliczu zidentyfikowanego problemu proponujemy stworzenie innowacyjnego systemu, który w znaczący sposób usprawni wczesną identyfikację i kontrolę potencjalnych ognisk epidemicznych. Projekt zakłada opracowanie zintegrowanej platformy, która umożliwi:

1. **Łatwe zgłaszanie objawów**: Użytkownicy będą mogli przesyłać informacje o swoich objawach za pośrednictwem intuicyjnego interfejsu obsługiwanego za pomocą języka naturalnego. Umożliwi to szybkie przeprowadzenie wstępnej oceny medycznej bez potrzeby bezpośredniego kontaktu z placówkami zdrowia, co zredukuje obciążenie systemu opieki zdrowotnej.

2. **Gromadzenie i analiza danych zdrowotnych**: Dane zebrane od użytkowników będą przechowywane i analizowane w czasie rzeczywistym, co pozwoli na identyfikację oraz śledzenie wzorców zachorowań w różnych lokalizacjach.
3. **Wsparcie dla służb zdrowia**: System zapewni cenne informacje instytucjom medycznym i badaczom, ułatwiając monitorowanie sytuacji epidemiologicznej oraz wspieranie decyzji dotyczących zarządzania zdrowiem publicznym.
4. **Informacje zwrotne i rekomendacje**: Użytkownicy otrzymają rekomendacje dotyczące dalszych kroków, np. czy powinni skonsultować się ze specjalistą, oraz informacje na temat lokalnych placówek medycznych.

Dzięki temu rozwiązaniu możliwe jest nie tylko usprawnienie indywidualnego zarządzania zdrowiem użytkowników, ale także wsparcie służb medycznych w reagowaniu na sytuacje kryzysowe związane z chorobami zakaźnymi. W rezultacie proponowany system stanowi istotny wkład w poprawę zdrowia publicznego i bezpieczeństwa epidemiologicznego.

1.4. Proponowana architektura rozwiązania

Planowana jest implementacja systemu wieloagentowego opartego na dużych modelach językowych (LLM), co umożliwi efektywne rozdzielenie zadań między wyspecjalizowanymi modułami, zwiększając możliwości przetwarzania i analizy danych. Dzięki zastosowaniu LLM, system będzie mógł lepiej rozumieć kontekst i intencje użytkowników, dostarczając trafniejsze rekomendacje oraz szybciej adaptując się do dynamicznie zmieniających się warunków.

Poniżej znajdują się opisy prototypowych agentów, które pełnią różne funkcje w systemie, począwszy od prowadzenia rozmów z użytkownikiem, aż po analizę i wizualizację danych dotyczących ognisk epidemicznych. Każdy z tych agentów specjalizuje się w konkretnym zadaniu, co zwiększa precyzję i skuteczność całego systemu w reagowaniu na potencjalne zagrożenia zdrowotne.

- Agent chatbota – prowadzi rozmowy z użytkownikiem, uwzględniając historię konwersacji.
- Agent walidacji – analizuje wiadomości użytkownika pod kątem medycznym, normalizuje listę objawów (w tym celu można zdefiniować listę dostępnych opcji) oraz informuje agenta chatbota, aby poprosił o lokalizację użytkownika, jeśli nie została podana wcześniej.
- Agent doradczy – określa właściwą dziedzinę medycyny (np. dermatologia, onkologia) i wykonuje podstawowe polecenia medyczne, które są następnie przekazywane do agenta chatbota w celu dalszej komunikacji z użytkownikiem. Przekazuje również informacje o powiązaniu lokalizacji z objawami do agenta analizy danych medycznych. Możliwe jest doradztwo bez opierania się na RAG, korzystając z wiedzy wbudowanej w LLM, w celu uproszczenia procesu implementacji.
- Agent analizy danych medycznych – zbiera dane dotyczące lokalizacji i objawów, identyfikuje powtarzające się wzorce w określonych obszarach, przekazuje zidentyfikowane przypadki w spójnych obszarach do agenta wizualizacyjnego oraz dostarcza informacji o potencjalnym zagrożeniu epidemicznym agentowi chatbota.
- Agent wizualizacyjny – wizualizuje ogniska na mapie, zaznaczając obszary o podobnych objawach w jednolitych kolorach.

1.4.1. Potencjalne rozszerzenie zakresu systemu:

- Agent prognozowania epidemii – analizuje wzrost obszaru epidemii w czasie, modelując rozwój epidemii w przyszłości, przekazuje te dane do agenta administratora, który je wizualizuje.
- Integracja z placówkami medycznymi, aby zbierać więcej informacji o przypadkach chorobowych w celu usprawnienia analizy i prognozowania ognisk.
- Sugerowanie użytkownikom odpowiednich placówek medycznych w pobliżu.

1.5. Udział w Hackathonie

Nasz zespół planuje wzięcie udziału w hackathonie, który odbywa się na platformie lablab.ai. Hackathon Challenge, pod hasłem Lōkahi, skupia się na współpracy i innowacjach w dziedzinie opieki zdrowotnej.

Celem wydarzenia jest stworzenie ekosystemu zdrowotnego, który integruje różne aspekty opieki medycznej, płatności, zdrowia publicznego, bezpieczeństwa pacjentów i badań klinicznych przy pomocy sztucznej inteligencji. Rozwiązania te mają oferować większą kontrolę prywatności, poprawiać zdrowie przy niższych kosztach oraz promować równość zdrowotną.

Hackathon jest otwarty do 24 listopada. Informacje o hackathonie są dostępne pod linkiem.

2. Raport B - wymagania, GAIA, BPMN

2.1. Wymagania

2.1.1. Wymagania funkcjonalne

- **Zgłaszanie objawów**: System powinien umożliwiać użytkownikom zgłaszanie objawów za pomocą intuicyjnego interfejsu.
- **Analiza danych zdrowotnych**: System musi gromadzić i analizować dane w czasie rzeczywistym.
- **Wsparcie dla służb zdrowia**: System powinien dostarczać informacji instytucjom medycznym i badaczom.
- **Rekomendacje dla użytkowników**: System powinien oferować użytkownikom rekomendacje dotyczące dalszych kroków.

2.1.2. Wymagania niefunkcjonalne

- **Skalowalność**: System musi być w stanie obsłużyć dużą liczbę użytkowników jednocześnie.
- **Bezpieczeństwo**: Dane użytkowników muszą być chronione zgodnie z obowiązującymi przepisami o ochronie danych.
- **Dostępność**: System powinien być dostępny 24/7.
- **Użyteczność**: Interfejs użytkownika powinien być prosty i intuicyjny.

2.1.3. Wymagania techniczne

- **Integracja z LLM**: System powinien wykorzystywać duże modele językowe do analizy i przetwarzania danych.
- **Architektura wieloagentowa**: System powinien być zbudowany w oparciu o architekturę wieloagentową, aby efektywnie rozdzielać zadania.

2.1.4. Wymagania dotyczące interesariuszy

- **Służby zdrowia**: System musi spełniać potrzeby instytucji medycznych i badaczy.
- **Użytkownicy końcowi**: System powinien być łatwy w użyciu dla pacjentów i lekarzy.

2.2. GAIA: faza analizy

2.2.1. Model ról

Role:

1. **Użytkownik (Rola: Interakcja z Chatbotem)**
   - **Opis:**

- Wchodzi w interakcję z agentem chatbota, pytając o porady medyczne.
- Udostępnia swoją lokalizację i inne istotne dane w celu uzyskania informacji na temat potencjalnych chorób lub zaleceń zdrowotnych.

- **Protokoły i aktywności:**
  - **Ask** - Przekazuje agentowi Chatbota swoje objawy i zebraną lokalizację.
  - **CollectLocalization** - Udostępnia swoją lokalizację.
  - **ReceiveMessage** - Otrzymuje i wyświetla wiadomość zwrotną od agenta Chatbota.
- **Obowiązki:**
  - **Liveness:**
    - **AskChatbot =** (CollectLocalization, Ask)
    - **ReceiveMessage =** (ReceiveMessage)
- **Bezpieczeństwo:**
  - Nie dotyczy.
- **Pozwolenia:**
  - **reads:** informacje od agenta chatbota
  - **generates:** zapytania do agenta chatbota

2. **Agent Chatbota (Rola: Prowadzący rozmowę)**
   - **Opis:**
     - Prowadzi rozmowy z użytkownikiem, uwzględniając historię konwersacji.
     - Zbiera lokalizację, czas i wiadomość użytkownika.
     - Zapewnia anonimizację danych, usuwając ewentualne wrażliwe informacje dostarczone przez użytkownika (np. PESEL).
     - Posiada zabezpieczenie przed wprowadzaniem przez użytkownika nonsensownych informacji.
     - Komunikuje się z innymi agentami w celu uzyskania informacji lub podjęcia działań.
   - **Protokoły i aktywności:**
     - **AnonymizeData** - Przetwarza i anonimizuje dane użytkownika w celu usunięcia wrażliwych informacji.
     - **ValidateUserInput** - Waliduje wprowadzone przez użytkownika dane pod kątem ich sensowności.
     - **RephraseRequest** - Parafrazuje zapytanie użytkownika przed wysłaniem do agentów wiedzy.
     - **InformKnowledgeAgents** - Przesyła sparafrazowane zapytanie do agentów Wiedzy w celu uzyskania informacji medycznych.
     - **AwaitAgregator** - Czeka na zebrane i zsumowane dane od agenta agregującego.
     - **InformAnalyzer** - Po otrzymaniu danych z agenta Agregującego, przekazuje lokalizację, czas, wiadomość użytkownika i zagregowane dane medyczne do agenta analizatora danych.
     - **RespondUser** - Po otrzymaniu danych od agenta agregującego, generuje wynikową odpowiedź z poleceniami dla użytkownika.
   - **Obowiązki:**
     - **Liveness:**
       - **GenerateResponse =** (AnonymizeData, ValidateUserInput, RephraseRequest, InformKnowledgeAgents, AwaitAgregator, InformDataAnalyzer, RespondUser)
   - **Bezpieczeństwo:**
     - Anonimizacja danych pozyskanych od użytkownika.
     - Przerwanie procesowania w razie wprowadzenia nonsensownej wiadomości.
   - **Pozwolenia:**
     - **reads:** wiadomość użytkownika
     - **changes:** *RephraseRequest* (parafrazowanie zapytania użytkownika)
     - **generates:** sparafrazowane zapytanie do agentów wiedzy
3. **Agenty Wiedzy (Rola: Eksperci wiedzy)**
   - **Opis:**
     - Przeszukują różne zdefiniowane źródła wiedzy medycznej.

    - Wyciągają dane medyczne na podstawie otrzymanego sparafrazowanego zapytania użytkownika.
  - **Protokoły i aktywności:**
    - **AwaitChatbot** - Oczekują na sparafrazowane zapytanie od agenta Chatbota.
    - **SearchMedicalKnowledge** - Przeszukują zewnętrzne skonfigurowane źródła wiedzy medycznej na podstawie otrzymanego zapytania.
    - **ProvideKnowledgeToAggregator** - Przekazują zebrane informacje medyczne do agenta agregującego.
  - **Obowiązki:**
    - **Liveness:**
      - **RespondToAggregator =** (AwaitChatbot, SearchMedicalKnowledge, ProvideKnowledgeToAggregator)
  - **Bezpieczeństwo:**
    - Nie dotyczy.
  - **Pozwolenia:**
    - **reads:** sparafrazowane zapytanie użytkownika
    - **generates:** surowe dane medyczne
4. **Agent Agregujący (Rola: Agregator danych)**
  - **Opis:**
    - Agreguje dane z różnych źródeł od agentów Wiedzy.
    - Tworzy spójną informację medyczną.
  - **Protokoły i aktywności:**
    - **AwaitKnowledgeAgents** - Czeka na informacje od agentów Wiedzy.
    - **AggregateData** - Analizuje i przetwarza zebrane dane, tworząc podsumowanie.
    - **InformChatbot** - Informuje agenta Chatbota o zagregowanych danych medycznych.
  - **Obowiązki:**
    - **Liveness:**
      - **RespondToChatbot =** (AwaitKnowledgeAgents, AggregateData, InformChatbot)
  - **Bezpieczeństwo:**
    - Nie dotyczy.
  - **Pozwolenia:**
    - **reads:** informacje od agentów wiedzy
    - **generates:** zagregowane dane medyczne
5. **Agent Analizatora Danych Medycznych (Rola: Analizator danych medycznych)**
  - **Opis:**
    - Otrzymuje od agenta Chatbota lokalizację, czas, wiadomość użytkownika i zagregowane dane medyczne.
    - Przechowuje dane w bazach danych.
  - **Protokoły i aktywności:**
    - **AwaitChatbot** - Oczekuje danych od agenta Chatbota.
    - **StoreDetailedData** - Zapisuje czas, lokalizację, wiadomość użytkownika i zagregowane dane medyczne do jednej bazy danych.
    - **ExtractMedicalInformation** - Co jakiś czas przetwarza zgromadzone dane, tworząc "wyciąg" zawierający listę objawów, potencjalny typ choroby i dziedzinę medycyny.
    - **StoreExtractedData** - Zapisuje "wyciągi" wraz z czasem i lokalizacją do innej bazy danych.
  - **Obowiązki:**
    - **Liveness:**
      - **ProcessData =** (AwaitChatbot, StoreDetailedData, ExtractMedicalInformation, StoreExtractedData)
  - **Bezpieczeństwo:**

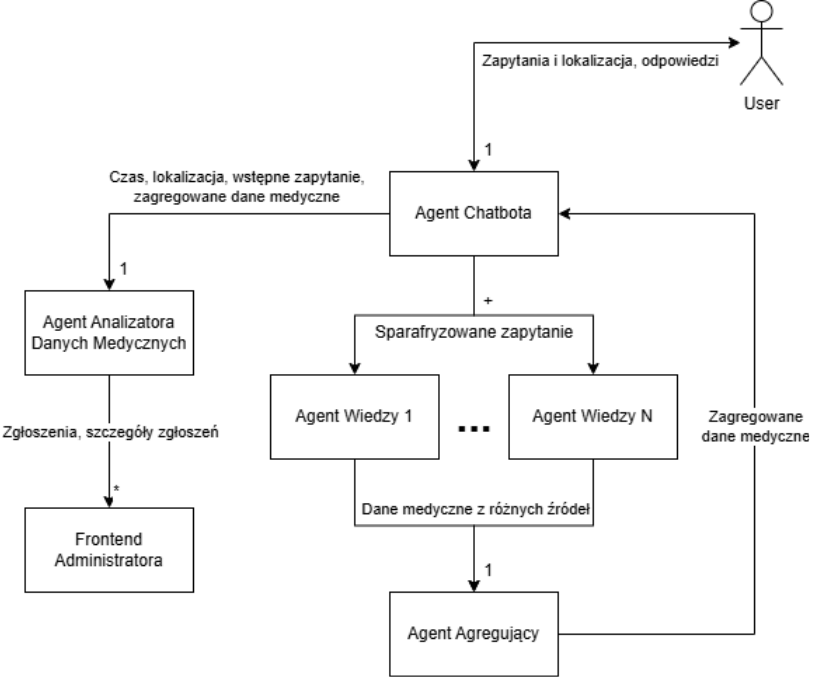
        - Zapewnia bezpieczne przechowywanie danych.
    - **Pozwolenia:**
        - **reads:** dane od agenta Chatbota (lokalizacja, czas, wiadomość użytkownika, zagregowane dane medyczne)
        - **changes:** przetwarzanie i wyciąganie informacji medycznych
        - **generates:** "wyciągi" z danych medycznych
6. **Frontend Administratora (Rola: Wizualizator danych)**
    - **Opis:**
        - Pobiera dane dotyczące zgłoszeń z API agenta Analizatora Danych Medycznych.
        - Wizualizuje zgłoszenia, podając "wyciągi", oraz umożliwia na żądanie pobranie początkowej wiadomości użytkownika i zagregowanych danych.
    - **Protokoły i aktywności:**
        - **FetchReportDataFromAPI** - Pobiera aktualne dane na temat zgłoszeń z API agenta Analizatora Danych Medycznych.
        - **DisplayReports** - Wyświetla zgłoszenia wraz z "wyciągami".
        - **RetrieveDetailedDataFromAPI** - Umożliwia administratorowi na żądanie pobranie początkowej wiadomości użytkownika i zagregowanych danych.
    - **Obowiązki:**
        - **Liveness:**
            - **UpdateReportVisualization** = (FetchReportDataFromAPI, DisplayReports)
            - **FetchDetailedData** = (RetrieveDetailedDataFromAPI)
    - **Bezpieczeństwo:**
        - Nie dotyczy.
    - **Pozwolenia:**
        - **reads:** dane z API agenta Analizatora Danych Medycznych

---

2.2.2. Model interakcji między rolami

- **Agent Chatbota ↔ Użytkownik:**
    - Prowadzi dialog z użytkownikiem.
    - Zbiera lokalizację, czas i wiadomość użytkownika.
- **Agent Chatbota → Agenty Wiedzy:**
    - Przekazuje sparafrazowane zapytanie do agentów Wiedzy.
- **Agenty Wiedzy → Agent Agregujący:**
    - Przekazują wyciągnięte dane medyczne do agenta Agregującego.
- **Agent Agregujący → Agent Chatbota:**
    - Przekazuje zagregowane dane medyczne do agenta Chatbota.
- **Agent Chatbota → Agent Analizatora Danych Medycznych:**
    - Przekazuje lokalizację, czas, wiadomość użytkownika i zagregowane dane medyczne.
- **Agent Analizatora Danych Medycznych → Frontend Administratora:**
    - Umożliwia pobieranie danych do wizualizacji zgłoszeń wraz z "wyciągami".
    - Umożliwia pobieranie na żądanie początkowej wiadomości użytkownika i zagregowanych danych.

2.3. GAIA: faza projektowania



2.3.1. Model agentów

- **Typy agentów i odpowiadające im role:**

Typ Agenta	Role	Liczba Instancji
Agent Chatbota	Prowadzący rozmowę	1
Agent Wiedzy	Ekspert wiedzy	1...N
Agent Agregujący	Agregator danych	1
Agent Analizatora Danych Medycznych	Analizator danych medycznych	1
Frontend Administratora	Wizualizator danych	*

- **Opis systemu:**
  - System składa się z pięciu typów agentów, z których trzy są pojedynczymi instancjami, jeden (Agent Wiedzy) posiada wiele instancji w zależności od liczby obsługiwanych źródeł wiedzy, a Frontend Administratora posiada wiele instancji, co umożliwia wielu użytkownikom pracę jednocześnie.

2.3.2. Model usług

Użytkownik

Usługa	Wejścia	Wyjścia	Warunki wstępne	Warunki końcowe
CollectLocalization	Lokalizacja (np. zgoda na dostęp do lokalizacji, ręczne wprowadzenie adresu)	Dane lokalizacji użytkownika	Loaklizacja nie została podana wcześniej	Lokalizacja użytkownika jest dostępna dla agenta Chatbota
Ask	Wiadomość tekstowa z opisem objawów, dolegliwości, pytań	Zapytanie przesłane do agenta Chatbota	Użytkownik podał swoją lokalizację Użytkownik zdefiniował zapytanie	Zapytanie użytkownika zostało przekazane agentowi Chatbota
ReceiveMessage	Wiadomość zwrotna od agenta Chatbota	Wyświetlona odpowiedź dla użytkownika	Użytkownik wysłał zapytanie do Chatbota i oczekuje na odpowiedź	Użytkownik otrzymał odpowiedź od agenta Chatbota

Agent Chatbota

Usługa	Wejścia	Wyjścia	Warunki wstępne	Warunki końcowe
AnonymizeData	Wiadomość użytkownika	Zanonimizowana wiadomość użytkownika	Wiadomość użytkownika jest dostępna	Wrażliwe dane zostały usunięte z wiadomości
ValidateUserInput	Zanonimizowana wiadomość użytkownika	Potwierdzenie poprawności lub informacja o błędzie	Zanonimizowana wiadomość użytkownika jest dostępna	Wiadomość zweryfikowana pod kątem sensowności lub proces przerwany w razie niesensowności
RephraseRequest	Walidowana wiadomość użytkownika	Sparafrazowane zapytanie	Wiadomość jest poprawna i sensowna	Sparafrazowane zapytanie zostało wygenerowane
InformKnowledgeAgents	Sparafrazowane zapytanie	Potwierdzenie wysłania do agentów Wiedzy	Sparafrazowane zapytanie jest dostępne	Zapytanie przekazane do agentów Wiedzy
AwaitAggregator	-	Zagregowane dane medyczne	Zapytanie wysłane do agentów Wiedzy	Otrzymano zagregowane dane medyczne od agenta Agregującego
InformDataAnalyzer	Lokalizacja, czas, wiadomość, dane medyczne	Potwierdzenie od agenta Analizatora Danych Medycznych	Otrzymano zagregowane dane medyczne	Dane przekazane do agenta Analizatora Danych Medycznych
RespondUser	Zagregowane dane medyczne	Odpowiedź dla użytkownika	Zagregowane dane medyczne są dostępne	Użytkownik otrzymał odpowiedź zawierającą informacje i polecenia

Agenty Wiedzy

Usługa	Wejścia	Wyjścia	Warunki wstępne	Warunki końcowe
AwaitChatbot	-	Sparafrazowane zapytanie od Chatbota	Agent oczekuje na zapytanie	Otrzymano zapytanie od agenta Chatbota
SearchMedicalKnowledge	Sparafrazowane zapytanie	Surowe dane medyczne	Otrzymano zapytanie	Dane medyczne zostały wyciągnięte z przypisanego źródła
ProvideKnowledgeToAggregator	Surowe dane medyczne	Potwierdzenie przekazania danych	Surowe dane medyczne są dostępne	Dane medyczne przekazane do agenta Agregującego

Agent Agregujący

Usługa	Wejścia	Wyjścia	Warunki wstępne	Warunki końcowe
AwaitKnowledgeAgents	-	Dane od agentów Wiedzy	Agent oczekuje na dane od agentów Wiedzy	Otrzymano dane od agentów Wiedzy
AggregateData	Dane od agentów Wiedzy	Zagregowane dane medyczne	Dane od agentów Wiedzy są kompletne	Dane zostały zagregowane i przeanalizowane
InformChatbot	Zagregowane dane medyczne	Potwierdzenie wysłania do Chatbota	Zagregowane dane medyczne są dostępne	Zagregowane dane przekazane do agenta Chatbota

Agent Analizatora Danych Medycznych

Usługa	Wejścia	Wyjścia	Warunki wstępne	Warunki końcowe
AwaitChatbot	-	Dane od agenta Chatbota	Agent oczekuje na dane od Chatbota	Otrzymano dane od agenta Chatbota
StoreDetailedData	Lokalizacja, czas, wiadomość, dane medyczne	Potwierdzenie zapisu danych	Dane od Chatbota są dostępne	Dane zapisane w bazie danych szczegółowych
ExtractMedicalInformation	Dane z bazy szczegółowej	Wyciągi z danych medycznych	Zgromadzone dane w bazie szczegółowej	Wyciągi wygenerowane z danych
StoreExtractedData	Wyciągi z danych medycznych	Potwierdzenie zapisu wyciągów	Wyciągi zostały wygenerowane	Wyciągi zapisane w bazie danych wyciągów
ProvideData	Żądanie od Frontendu Administratora	Dane zgłoszeń i wyciągów	Frontend zgłasza żądanie dostępu do danych	Dane zostały udostępnione Frontendowi Administratora
ProvideDetailedData	Żądanie od Frontendu Administratora	Zapytanie wejściowe użytkownika i zagregowane dane medyczne	Frontend zgłasza żądanie dostępu do szczegółowych danych	Szczegółowe dane zostały udostępnione Frontendowi Administratora

Frontend Administratora

Usługa	Wejścia	Wyjścia	Warunki wstępne	Warunki końcowe
FetchReportDataFromAPI	-	Dane zgłoszeń i wyciągów z API	Frontend jest uruchomiony i ma dostęp do API	Otrzymano aktualne dane z API
DisplayReports	Dane zgłoszeń i wyciągów	Wizualizacja zgłoszeń	Dane zostały pobrane z API	Zgłoszenia wraz z wyciągami wyświetlone dla administratora
RetrieveDetailedDataFromAPI	Żądanie administratora	Szczegółowe dane (wiadomość użytkownika, zagregowane dane)	Administrator zażądał szczegółowych danych	Szczegółowe dane zostały pobrane z API i wyświetlone na żądanie

---

2.3.3. Model znajomości

- **Użytkownik**
  - **Zna:**
    - Agenta Chatbota
- **Agent Chatbota:**
  - **Zna:**
    - Użytkownika
    - Wszystkich agentów Wiedzy.
    - Agenta Agregującego.
    - Agenta Analizatora Danych Medycznych.
- **Agenty Wiedzy:**
  - **Znają:**
    - Agenta Chatbota.
    - Agenta Agregującego.
- **Agent Agregujący:**

- **Zna:**
  - Wszystkich agentów Wiedzy.
  - Agenta Chatbota.
- **Agent Analizatora Danych Medycznych:**
  - **Zna:**
    - Agenta Chatbota.
- **Frontend Administratora:**
  - **Zna:**
    - Agenta Analizatora Danych Medycznych.

2.4. Schemat BPMN