PRISM
Project Group #1

Mahmut Osmanovic Isac Paulsson Sebastian Tuura
Clement Pickel Celian Friedrich

January 12, 2025

Contents

1 Introduction

Clinicians all around the world prescribe medications to patients on a daily basis. Unfortunately, these prescriptions are not without errors. Quantifying the error is difficult since there are a multitude of factors that contribute to it, and the strength of each contributing factor varies according to geographical region. Nonetheless, a recent study highlights that among hospitalized patients worldwide, 3–16% suffer from an injury as a result of medical intervention, the most common being the adverse effects of drugs [4]. Another study specifies that nearly 7% of prescriptions lead to harmful interactions [6]. It is beyond reason to expect clinicians to be aware of all possible significant medical interactions between all medical drugs. Especially as new drugs are continuously being released to the market.

We have developed PRISM (Predictive and Interactive System for Medications) in order to aid the medical professionals in their decision making. That includes general questions answering, side effects identification, alternative medicine suggestions and medication recommendation given symptoms. The application combines between traditional but reliable technologies and cutting edge tools. The main utilities include but are not limited to large language models (LLMs) and the automatization of SPARQL queries to the WikiData knowledge graph. The application is primarily intended to be utilized by medical professionals however may simultaneously be of high utility for the general populous.

2 Application Description

2.1 Application System

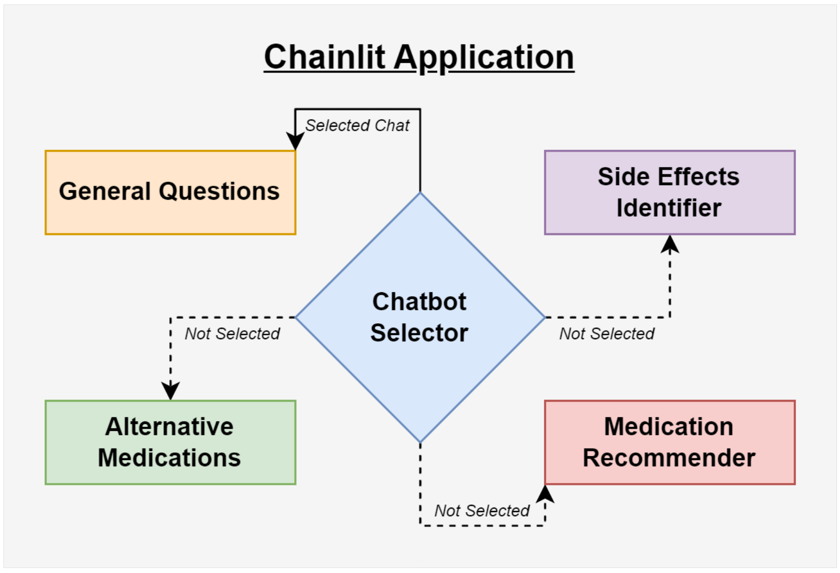


Figure 1: Figure highlights the application architecture from a bird's-eye view

The application consists of four different chat profiles with their own specific use cases, the chat profiles being:

Feature	Description
General Questions	Allows the user to query the LLM about general and health-related questions.
Side Effects Identifier	Provides whether there are any significant chemical interactions between the user's current list of medications and a proposed medication.
Alternative Medications	Suggests safe alternatives given a proposed medication based on the user's current medications.
Medication Recommender	Provides a list of medications tailored to address the user-specified symptoms.

Table 1: Description of Features

2.1.1 Shared Functionalities

Welcome Message and Instructions. A welcome message is initially presented to the user based on the selected profile. The welcome message includes a brief explanation of the profile as well as the expected input format.

Filtering for Medicine(s) and Symptom(s). User input is filtered using both a language model (LLM) and the FuzzyWuzzy library. User input is passed with a prompt to the LLM in order to handle malformed input. It is instructed to try to fix minor misspellings and reformat the input as requested. From this, a list is extracted and thereafter passed to FuzzyWuzzy in order to match with entries from WikiData, collected using SPARQL. This ensures that the retrieved medication names match the entries in WikiData. WikiData frequently expresses symptom descriptions in Latin. An additional LLM prompt is executed to translate the input to Latin. Symptoms are thus provided in both English and Latin to FuzzyWuzzy. Subsequently, the best match is returned as the matched medicine(s) or symptom(s).

Query Answer. Given the provided input, SPARQL queries are created based on the selected profile to retrieve the necessary data to answer the user query accurately. The data is provided in a prompt to the LLM together with instructions for how to format the data based on the users query. Results are thereafter displayed to the user.

2.1.2 General Questions Chat

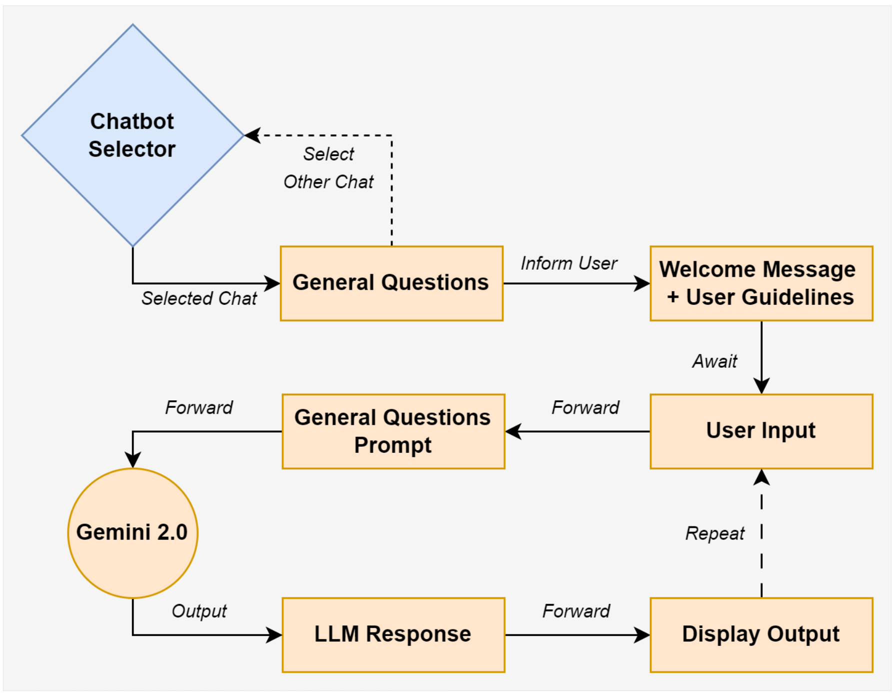


Figure 2: Figure highlights architectural design for General Questions

The general questions chat is the default chat. It generates answer to general user specified and healthcare related questions. After an initial welcome message with user guidelines, the user input is toggled through a meticulously written and tested LLM-Prompt and subsequently forwarded to the Gemini 2.0 LLM. The application awaits the Gemini 2.0 response and displays it. Thereafter the user has the option to *ask another question* or *inquire for further details* since the current chat context is not lost in between inputs.

2.1.3 Side Effects Identifier Chat

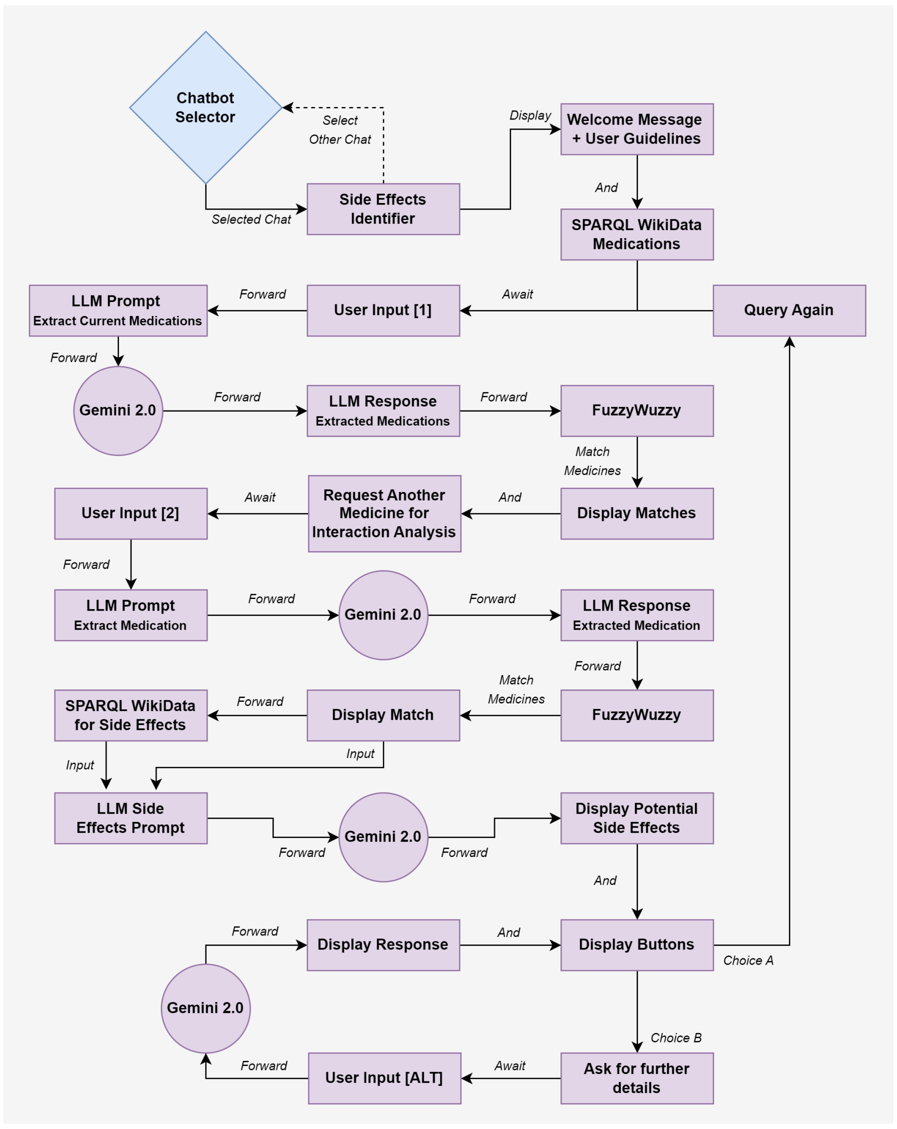
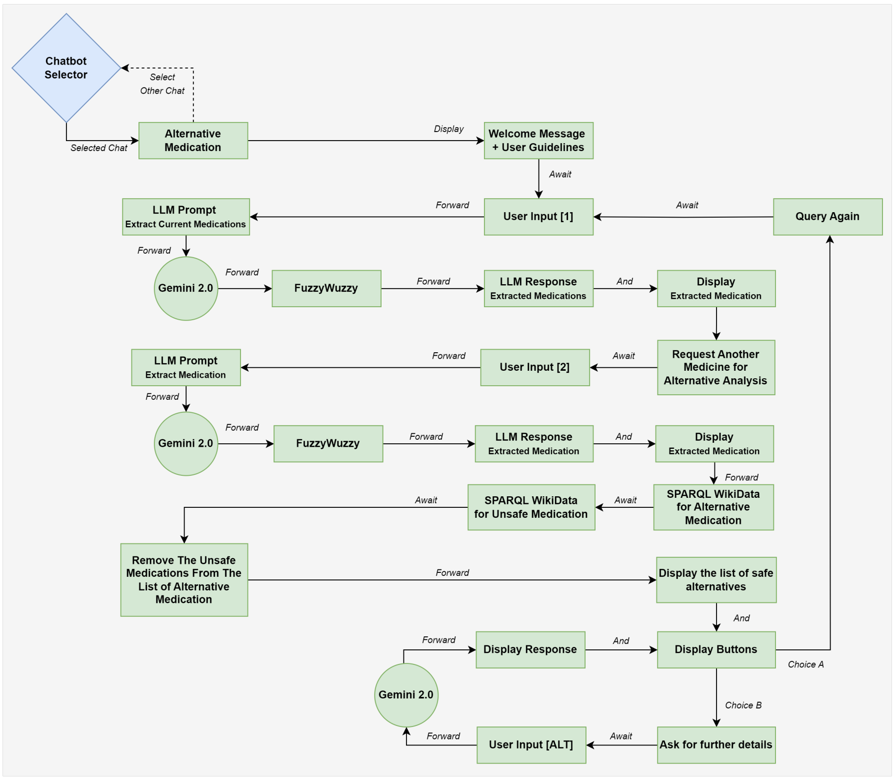


Figure 3: Figure highlights architectural design for General Questions

Figure [3] details the architectural design of the side effects identifier chat. The user inputs the current list of medications and the new medication. The chat returns whether or not there are any significant negative interactions between the current list of medications and the new, to be potentially used medicine. If there are any significant side effects, they are displayed in the output. Note that the available medications are queried at the start of the application. This ensures that the application stays up to date with the most recently released medical compounds, upon failure defaults to the most recently queried list of medications. The chat is designed with a high prioritization of reliability. Each input is first filtered through Gemini 2.0, which in accordance to a specified prompt, extracts the output of interest. The output is subsequently matched and scored, with FuzzyWuzzy, in accordance to its match to the queried and retrieved medications from WikiData. Finally the user is prompted with two buttons, either ask for further details within the given context, or start a new query, resetting the current context.

2.1.4 Alternative Medications Chat



Chatbot Selector
Select Other Chat
Selected Chat
Alternative Medication
Display
Welcome Message + User Guidelines
Await
LLM Prompt Extract Current Medications
Forward
User Input [1]
Await
Query Again
Gemini 2.0
FuzzyWuzzy
LLM Response Extracted Medications
And
Display Extracted Medication
LLM Prompt Extract Medication
User Input [2]
Request Another Medicine for Alternative Analysis
LLM Response Extracted Medication
SPARQL WikiData for Unsafe Medication
SPARQL WikiData for Alternative Medication
Remove The Unsafe Medications From The List of Alternative Medication
Display the list of safe alternatives
Display Response
Display Buttons
Choice A
Choice B
Ask for further details
User Input [ALT]

Figure 4: Figure specifies the architectural design for Alternative Medications

The alternative medications chat offers an alternative but equivalent medication to the proposed one, but without side effects. Similarly to the Side Effects Identifier Chat [4], it firstly extract the patients current list of medicine using an LLM prompt, Gemini 2.0 and FuzzyWuzzy, before displaying the matched side effects. It similarly extracts the new medication. Subsequently, safe alternative medicines are SPARQL queried from WikiData and displayed. Thereafter two buttons are displayed, one allowing the user to ask follow up questions and another to query again.

2.1.5 Medication Recommendation Chat

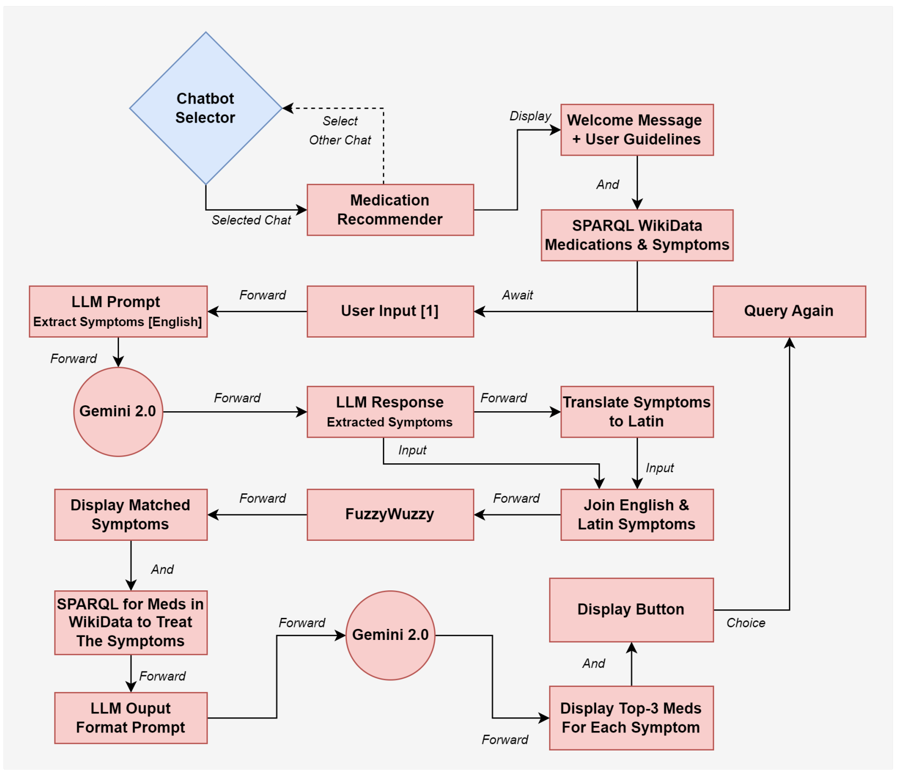


Figure 5: Figure highlights architectural design for Medication Recommendation

The Medication Recommendation chat in figure [5] allows a user to query for suitable medications based on experienced symptoms. The application queries available medications and symptoms from WikiData on startup. The user subsequently specifies their experienced symptoms (in english). They are translated to Latin because many symptoms in the knowledge graphs are specified in Latin. The extracted symptoms are matched against the queried list of symptoms from WikiData. Thereafter, they sorted and ranked, using FuzzyWuzzy, in accordence to the closest matching string. The matched symptoms are subsequently used to query WikiData for medications used to treat the symptoms. All of the queried data are toggled through a last prompt, specifying the desired output format for

Gemini 2.0. The output includes the top three most common and legal medications used to treat each of the symptoms. Lastly, a button is displayed that enables the user to make a new query.

2.2 Technical Details

2.2.1 Knowledge Graph

The knowledge graph used to query medical data from was WikiData. As of mid-2024, WikiData had 1.57 billion semantic triples. WikiData is a free and collaborative knowledge base hosted by the Wikimedia Foundation and is the common source for open data for WikiMedia projects like Wikipedia. Each item in WikiData is allocated a unique and positive integer prefixed with the upper-case letter Q, named QID. This means that item labels do not have to be unique, rather only their QID has to be unique.

WikiData records information about any item through so called "Statements". It consists of key-value pairs which match a property with one or more entity values. The the following informal English sentence "Zopiclone is used to treat sleep disorder" would on WikiData be encoded in the following statement with property: *medical condition treated* (*P2175*) with the value: *sleep disorder* (*Q177190*) under the item: *zopiclone* (*Q220426*).

2.2.2 SPARQL

The application utilizes a total of nine SPARQL queries, created using the wrapper library *SPARQLWrapper* for Python. The SPARQL queries provides the application with several functionalities. They retrieve the WikiData QID by using the medication name. Moreover, the queries retrieve a list of all active substances available on WikiData. We can query symptoms by their QID and retrieve all corresponding symptoms available on WikiData. It is also possible to query side effects by their name, check for interactions between two medications and find alternative medications for given a specified medicine. To view the implementation of the SPARQL queries, please check out the **sparql.py** file found within the **src** subfolder. Examine query [4] for a representative example.

2.2.3 FuzzyWuzzy

FuzzyWuzzy is a Python library used for string matching. We match the user input against all available medications on Wikidata to ensure that we are able to provide the user with a reliable answer. FuzzyWuzzy uses Levenshtein Distance to calculate the difference between two strings. Levenshtein Distance [5] measures the difference between two sequences by counting the number of edits needed to make in order to change one sequence into the other. Which is the reason as to why it is referred to as Edit Distance. As an example, changing the string *hello* to *goodbye* requires seven edits and therefore has the Levenshtein Distance of seven. Whereas changing the string *hello* to *help* only has a Levenshtein Distance of two making it a closer match.

2.2.4 Large Language Model

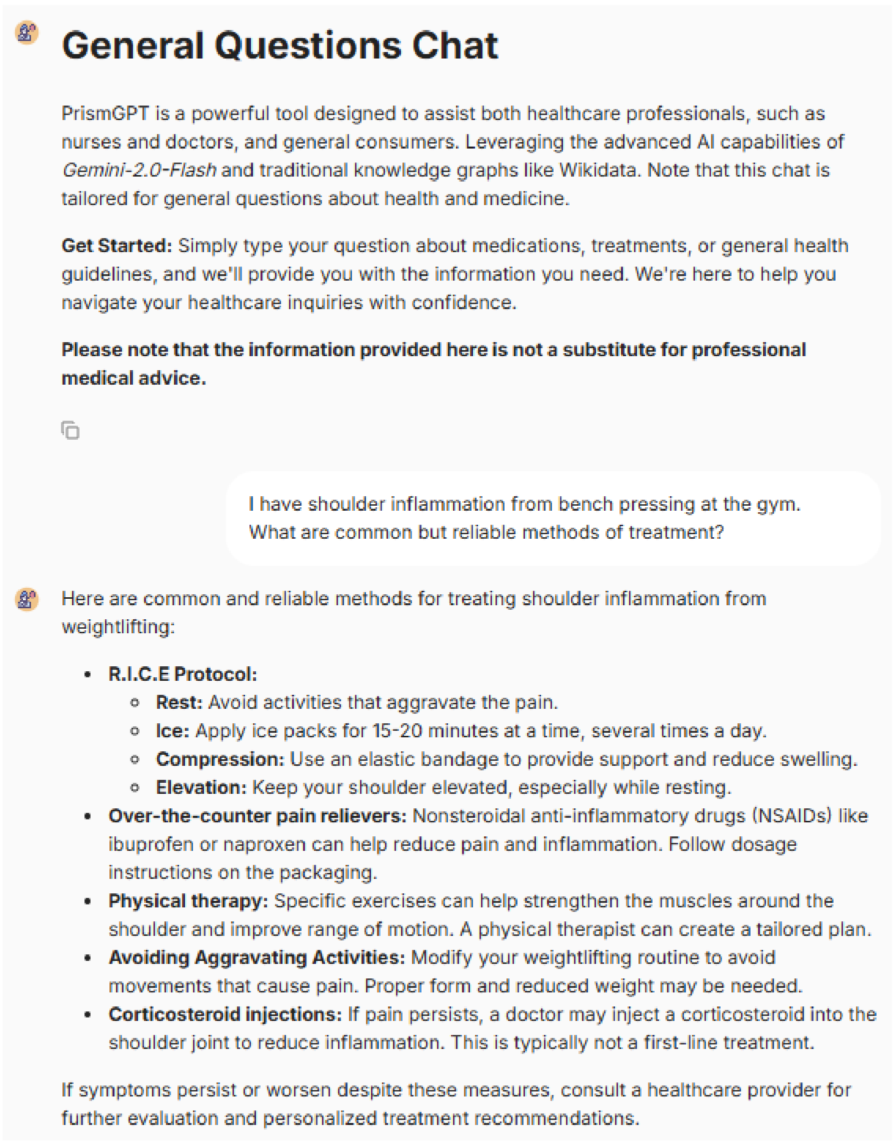
A variety of models where considered, including *meta-llama/llama-3.2-1B*, *meta-llama/llama-3.2-8B*, *EleutherAI/phi3:3.8b* and lastly *Gemini-2.0-Flash*. Several metrics were considered upon evaluating the model performance. We examined each models ability to parse, format user input and their ability of retaining the most important information in-between messages.

2.2.5 User Interface

In order to implement a conversational user interface for PRISM the python library Chainlit has been used. It provides an abundance of functionality and allows a simplified implementation of the interface. The interface looks good and is similar to many other prominent AI chats. Using profiles we have been able to separate the different functionalities of PRISM in to their own chats, this provides clarity and ease of use.

2.3 Application Usage

2.3.1 Use Case 1 | General Questions Chat



General Questions Chat

PrismGPT is a powerful tool designed to assist both healthcare professionals, such as nurses and doctors, and general consumers. Leveraging the advanced AI capabilities of *Gemini-2.0-Flash* and traditional knowledge graphs like Wikidata. Note that this chat is tailored for general questions about health and medicine.

Get Started: Simply type your question about medications, treatments, or general health guidelines, and we'll provide you with the information you need. We're here to help you navigate your healthcare inquiries with confidence.

Please note that the information provided here is not a substitute for professional medical advice.

I have shoulder inflammation from bench pressing at the gym. What are common but reliable methods of treatment?

Here are common and reliable methods for treating shoulder inflammation from weightlifting:

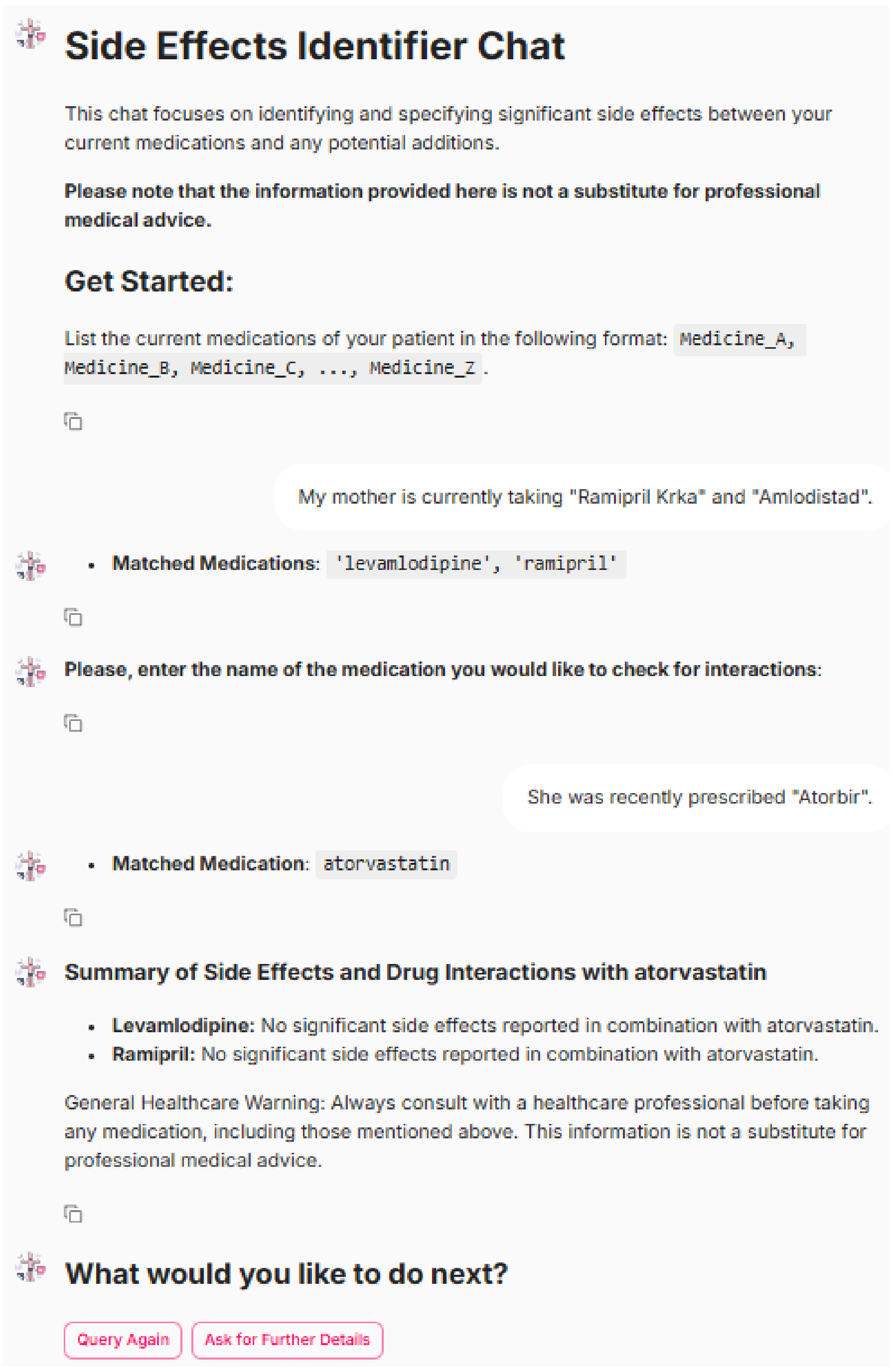
- **R.I.C.E Protocol:**
  - **Rest:** Avoid activities that aggravate the pain.
  - **Ice:** Apply ice packs for 15-20 minutes at a time, several times a day.
  - **Compression:** Use an elastic bandage to provide support and reduce swelling.
  - **Elevation:** Keep your shoulder elevated, especially while resting.
- **Over-the-counter pain relievers:** Nonsteroidal anti-inflammatory drugs (NSAIDs) like ibuprofen or naproxen can help reduce pain and inflammation. Follow dosage instructions on the packaging.
- **Physical therapy:** Specific exercises can help strengthen the muscles around the shoulder and improve range of motion. A physical therapist can create a tailored plan.
- **Avoiding Aggravating Activities:** Modify your weightlifting routine to avoid movements that cause pain. Proper form and reduced weight may be needed.
- **Corticosteroid injections:** If pain persists, a doctor may inject a corticosteroid into the shoulder joint to reduce inflammation. This is typically not a first-line treatment.

If symptoms persist or worsen despite these measures, consult a healthcare provider for further evaluation and personalized treatment recommendations.

Figure 6: A typical use case of the General Questions Chat.

The first scenario [6] highlights the use case of the General Questions Chat. The user receives guidelines on common treatments for shoulder inflammation.

2.3.2 Use Case 2 | Side Effects Identifier Chat



Side Effects Identifier Chat

This chat focuses on identifying and specifying significant side effects between your current medications and any potential additions.

Please note that the information provided here is not a substitute for professional medical advice.

Get Started:

List the current medications of your patient in the following format: `Medicine_A, Medicine_B, Medicine_C, ..., Medicine_Z`.

My mother is currently taking "Ramipril Krka" and "Amlodistad".

- **Matched Medications**: `'levamlodipine', 'ramipril'`

Please, enter the name of the medication you would like to check for interactions:

She was recently prescribed "Atorbir".

- **Matched Medication**: `atorvastatin`

Summary of Side Effects and Drug Interactions with atorvastatin

- **Levamlodipine:** No significant side effects reported in combination with atorvastatin.
- **Ramipril:** No significant side effects reported in combination with atorvastatin.

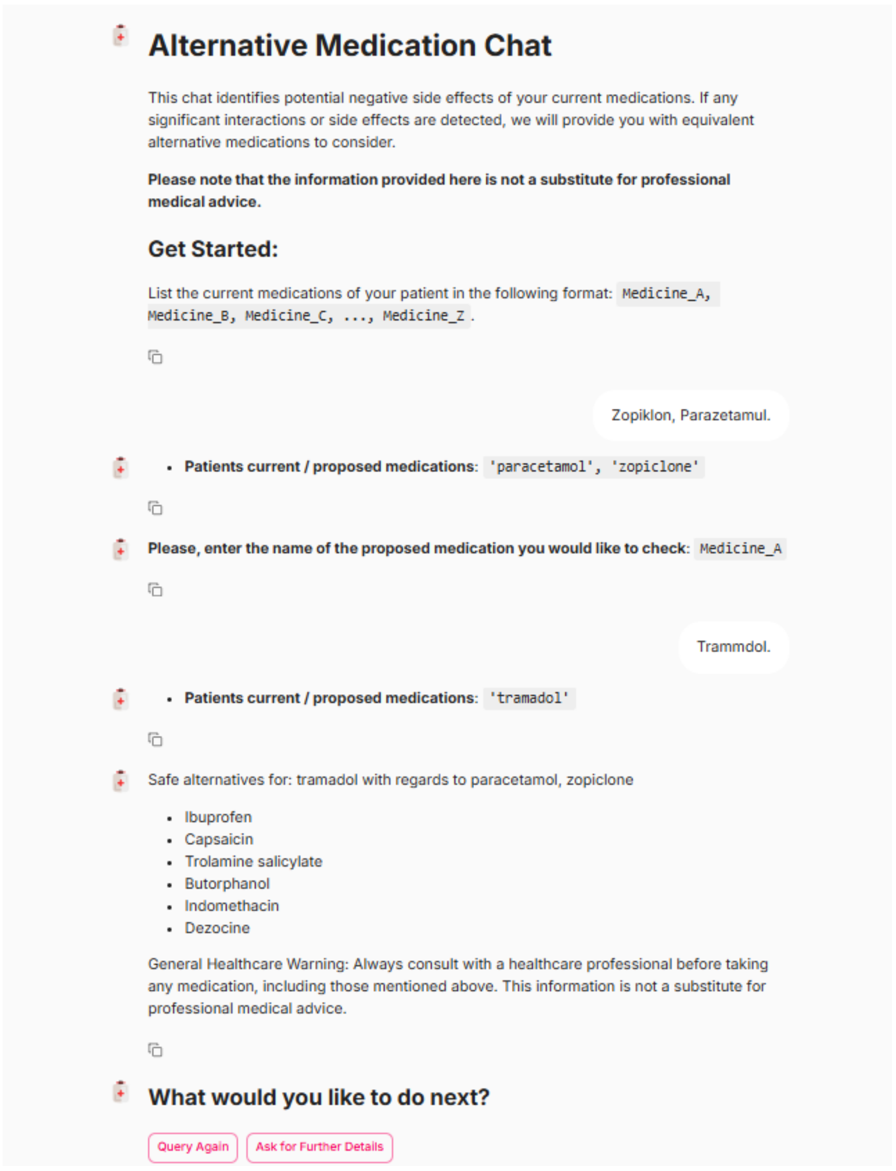
General Healthcare Warning: Always consult with a healthcare professional before taking any medication, including those mentioned above. This information is not a substitute for professional medical advice.

What would you like to do next?

Figure 7: A typical use case of the Side Effects Identifier Chat.

The second scenario [7] is a real use case of the Side Effects Identifier Chat. It manages to correctly identify the active substances in each of the entered medications [1] [2] [3]. This occurs even though brand-names were entered (in place of active substances) and that the expected user input was deviated from.

2.3.3 Use Case 3 | Alternative Medications Chat



Alternative Medication Chat

This chat identifies potential negative side effects of your current medications. If any significant interactions or side effects are detected, we will provide you with equivalent alternative medications to consider.

Please note that the information provided here is not a substitute for professional medical advice.

Get Started:

List the current medications of your patient in the following format: `Medicine_A, Medicine_B, Medicine_C, ..., Medicine_Z`.

Zopiklon, Parazetamul.

- **Patients current / proposed medications:** `'paracetamol', 'zopiclone'`

Please, enter the name of the proposed medication you would like to check: `Medicine_A`

Trammdol.

- **Patients current / proposed medications:** `'tramadol'`

Safe alternatives for: tramadol with regards to paracetamol, zopiclone

- Ibuprofen
- Capsaicin
- Trolamine salicylate
- Butorphanol
- Indomethacin
- Dezocine

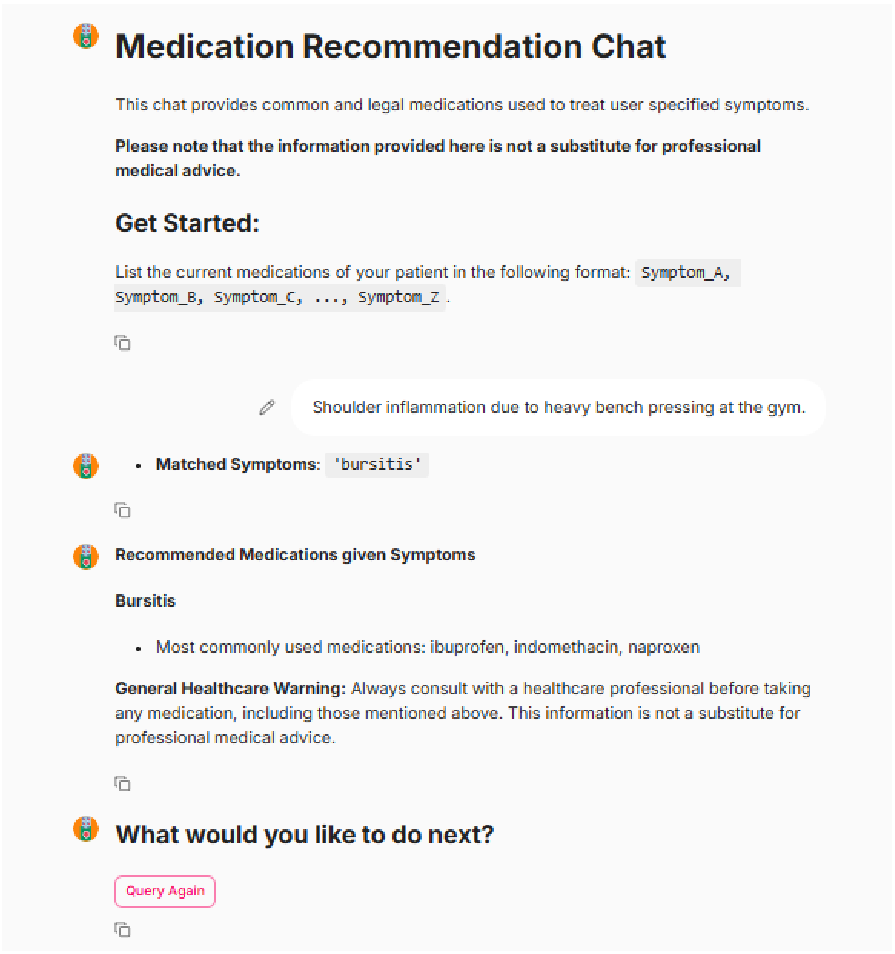
General Healthcare Warning: Always consult with a healthcare professional before taking any medication, including those mentioned above. This information is not a substitute for professional medical advice.

What would you like to do next?

Figure 8: A typical use case of the Alternative Medications Chat.

The third highlights a use case [8] in which there exists a negative significant negative side effect between *paracetamol* and *zopiclone*. A close substitute for Paracetamol is suggested, namely *Ibuprofen*. In addition, make note of that the Alternative Medication Chat manages to correctly extract incorrectly spelled input medication names.

2.3.4 Use Case 4 | Medication Recommendation Chat



Medication Recommendation Chat

This chat provides common and legal medications used to treat user specified symptoms.

Please note that the information provided here is not a substitute for professional medical advice.

Get Started:

List the current medications of your patient in the following format: `Symptom_A, Symptom_B, Symptom_C, ..., Symptom_Z`.

Shoulder inflammation due to heavy bench pressing at the gym.

- **Matched Symptoms:** `'bursitis'`

Recommended Medications given Symptoms

Bursitis

- Most commonly used medications: ibuprofen, indomethacin, naproxen

General Healthcare Warning: Always consult with a healthcare professional before taking any medication, including those mentioned above. This information is not a substitute for professional medical advice.

What would you like to do next?

Figure 9: A typical use case of the Medication Recommendation Chat.

The final use case [9] manages to retrieve commonly used medications given the user specified input. Note that the chat manages to correctly match and retrieve *bursitis*, which is the Latin and medical term for *shoulder inflammation*.

3 Discussion

The choice of *WikiData* over *OBO Foundry* is based on two main reasons. Firstly, all application necessary information was not available at *OBO Foundry.* Secondly, *OBO Foundry* was less intuitive to query from.

Pinecone is a vector database which we considered as an alternative for string matching. By embedding the strings as vectors (through DistilBERT), one can effectively compare their likeness through the use of cosine similarity. Nonetheless FuzzyWuzzy clearly outperformed this method for small string inputs. For example, it misclassified *zopiclone* if even a single letter was omitted, i.e., *zopiclon.* The matching with Pinecone may probably be improved by using other embedding methods or larger string inputs.

The small meta-llama model *llama-3.2-1B* was the smallest one tested. It is sometimes unwilling to answer questions even when explicitly prompted. In addition, it lacked the ability to remember conversational context due to having a small context window.*llama-3.2-8B* outperforms *llama-3.2-1B* substantially. Its primary issue is the execution time on less powerful hardware, which leads to large variability in run time between computers. Through further exploration of available models we found that Google's state of the art Gemini Model had a free tier API endpoint. Gemini was chosen as our language model since it addressed many of the aforementioned issues. It is fast at parsing user information with high accuracy, able to remember the conversational context and linguistically expressive. In order to provide a interactive user experience, it is of vital importance that our language model is able to remember the conversational context.

Fabricated or incorrect responses are a common occurrence in LLMs. Through the usage of SPARQL queries we enable the LLM to provide fact-based responses derived directly from a structured knowledge base like WikiData. Integrating LLMs and SPARQL queries enables the LLM to retrieve specific information dynamically rather than solely relying on pre-trained knowledge, making it valuable in fields where precision is critical, such as in the medical industry.

The usage of Chainlit allowed us to focus on the development of the application and whilst retaining a intuitive and aesthetically pleasing user interface. Chainlit's simplicity of use was of great value.

Ethical concerns related to the utilization of LLMs in the Healthcare Industry includes patient privacy and model transparency. PRISM does not require an patient sensitive data in order to function properly. By querying WikiData we can effectively circumvent issues with model transparency.

4 Appendix

SPARQL query for finding alternative medications that are used to treat the same conditions as the medication being queried:

```
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>

SELECT DISTINCT ?alternative ?alternativeLabel
WHERE {{
# P2175 referes to "Medical condition treated" on Wikidata
  {medB} wdt:P2175 ?disorder.
  ?alternative wdt:P2175 ?disorder.
  FILTER(?alternative != {medB})
  SERVICE wikibase:label {{
    bd:serviceParam wikibase:language "en".
  }}
}} LIMIT 15
```

References

[1] FASS.se. Amlodistad. https://www.fass.se/LIF/product?userType=2&nplId=20131031000098, . Accessed: 2025-01-12; FASS is a national medical information system (NMI). NMI ID for the FASS platform: 23.7.

[2] FASS.se. Atorbir. https://www.fass.se/LIF/product?userType=2&nplId=20091211000037, . Accessed: 2025-01-12; FASS is a national medical information system (NMI). NMI ID for the FASS platform: 23.7.

[3] FASS.se. Ramipril krka. https://www.fass.se/LIF/product?userType=0&nplId=20100615000038, . Accessed: 2025-01-12; FASS is a national medical information system (NMI). NMI ID for the FASS platform: 23.7.

[4] F Oyebode. Clinical errors and medical negligence. *Med Princ Pract*, 22(4): 323–333, 2013. doi: 10.1159/000346296.

[5] Dan Jurafsky Stanford University, CS124 Lecture. Minimum edit distance. https://web.stanford.edu/class/cs124/lec/med.pdf. Accessed: 2025-01-12.

[6] RA Tariq, R Vashisht, A Sinha, and Y Scherbak. Medication dispensing errors and prevention. *StatPearls [Internet]*, 2025.