

# Data specification navigator (user documentation)

## 1 Introduction

This document is a user guide for the Data specification navigator application. It is intended for end-users who wish to navigate and query data specifications using simple, natural language.

The Data Specification Navigator is a conversational tool designed to help you interact with your data specification. It allows you to ask questions about your data specification in plain English and, in return, provides you with the correct technical query (SPARQL) to get the data you need.

### 1.1 Target audience

This document's target audience are end-users who want to start using the application. It is assumed that the application has been deployed and is accessible to the user.

## 2 Getting started

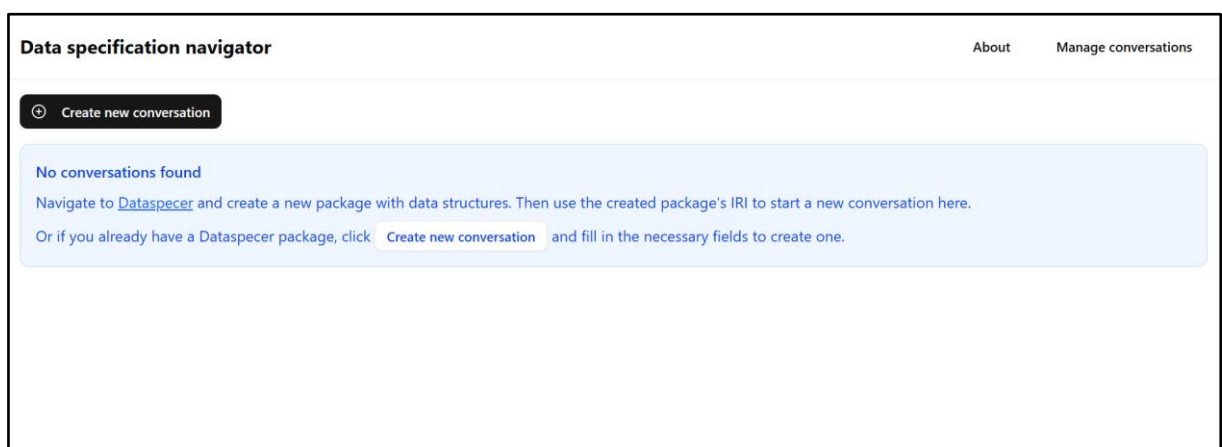
This application works directly with data specifications created in the Dataspecer tool therefore, some knowledge about Dataspecer is required. If you don't know how to use Dataspecer, please consult its documentation and guides: <https://dataspecer.com/>

### 2.1 Workflow

This section demonstrates the workflow with this application on a concrete example with screenshots for better visualization. The important UI elements are discussed in more detail in chapter 3).

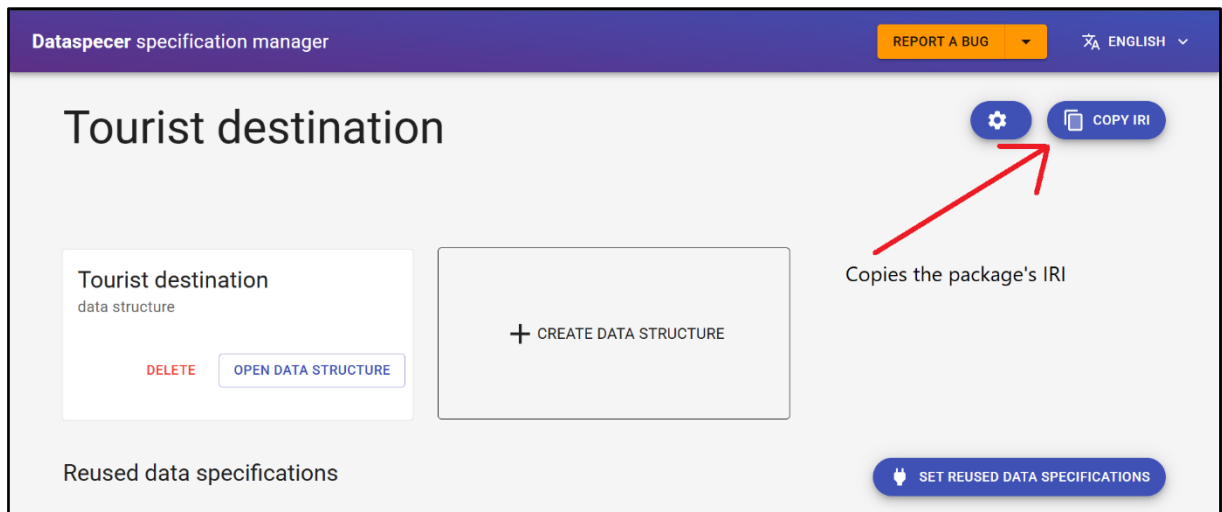
The user navigates to the main page of the Data specification navigator, which is by default the conversation management view (see Figure 1).

Figure 1: No conversations created yet



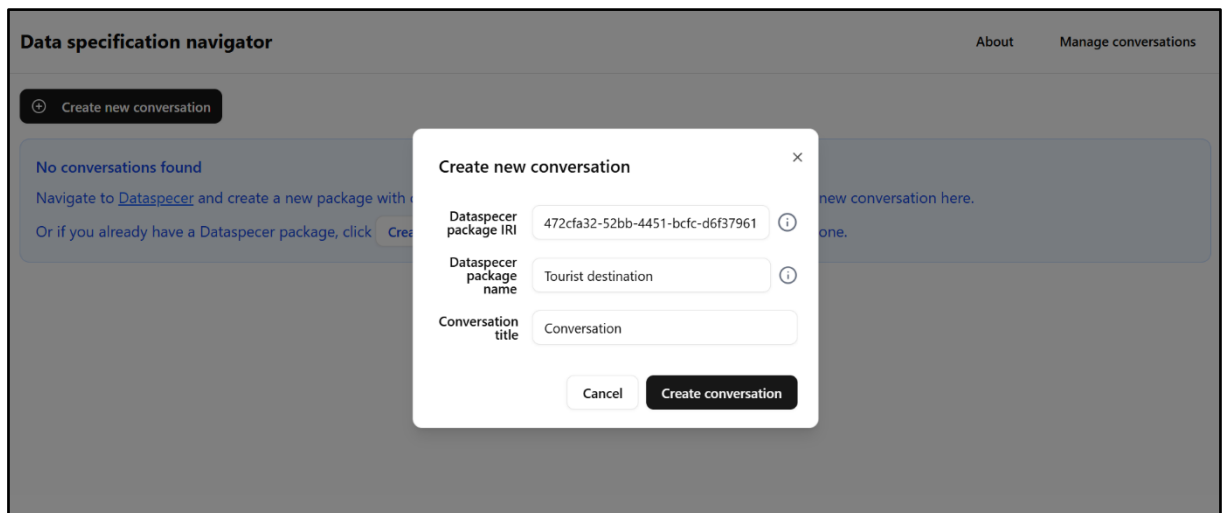
The user follows the instructions and creates a new Dataspecer package. The user then copies the package IRI (see Figure 2).

Figure 2: Dataspecer package UI



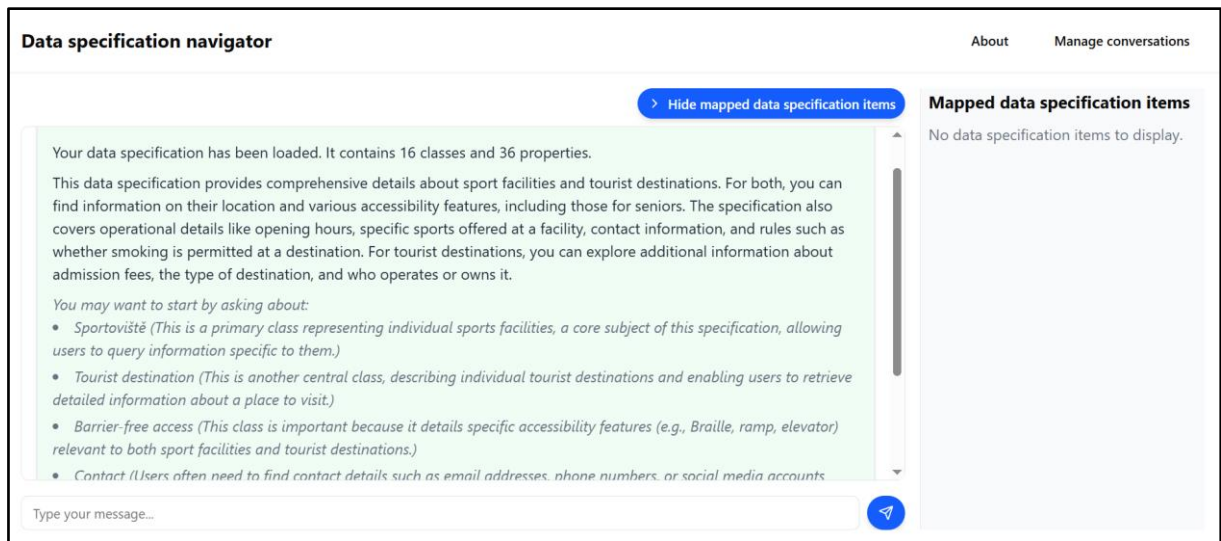
The user returns to the Data specification navigator, clicks on the “Create new conversation” and a form appears. The user fills in the necessary information (see Figure 3).

Figure 3: Conversation creation form



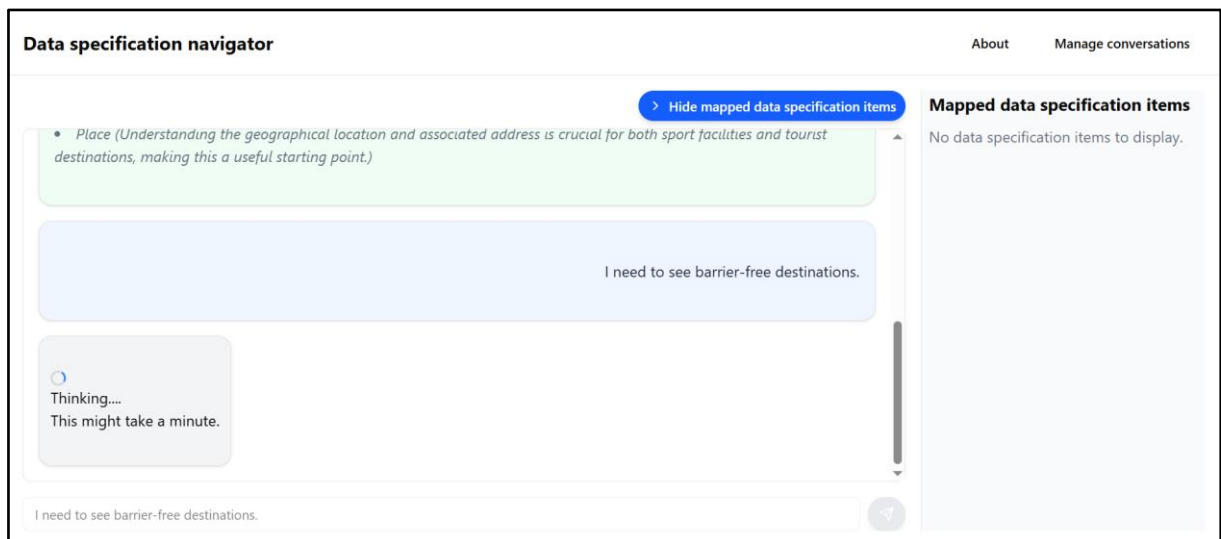
The user opens the newly created conversation and sees the welcome message (see Figure 4).

Figure 4: Welcome message



The user types their first message and sends it (see Figure 5).

Figure 5: Waiting for a chatbot reply



The chatbot sends an answer back. The right-side panel is updated with mapped items from the data specification. The reply from the chatbot is long so it is shown in two parts (see Figure 6 and Figure 7).

Figure 6: Chatbot reply - SPARQL query

The interface is titled "Data specification navigator" and includes links for "About" and "Manage conversations". A toggle button "Hide mapped data specification items" is present. The chatbot message states: "I have processed your message and found some relevant information." Below this, the SPARQL query is displayed in a code block with a "Copy" icon:

```
SELECT DISTINCT *
WHERE {
  # Tourist destination
  ?Tourist_destination a <https://slovník.gov.cz/datový/turistické-cíle/pojem/turistický-cíl> .
  ?Tourist_destination <https://slovník.gov.cz/datový/turistické-cíle/pojem/bezbariérovost>
  ?Barrier-free_access .
  # Barrier-free access
  ?Barrier-free_access a <https://slovník.gov.cz/generický/bezbariérové-přístupy/pojem/bezbariérový-přístup> .
```

Below the query is a "Show your message" button and a text input field "Type your message...". On the right, the "Mapped data specification items" section lists:

- Tourist destination** (<https://slovník.gov.cz/datový/turistické-cíle/pojem/turistický-cíl>)  
Object properties:
  - **bezbariérovost** (→ Barrier-free access)
- Barrier-free access** (<https://slovník.gov.cz/generický/bezbariérové-přístupy/pojem/bezbariérový-přístup>)

Figure 7: Chatbot reply - suggestions

The interface is the same as Figure 6, but the chatbot message now says: "You might want to add to your message:". Below this, suggestions are listed under two categories:

**Tourist destination:**

- ☐ → [has touristic destination type](#) → [Touristic destination type](#)
- ☐ → [kontakt](#) → [Contact](#)
- ☐ → [otevřací doba](#) → [Time specification](#)
- ☐ → [vstupné](#) → [Admission](#)
- ☐ → [umístění](#) → [Place](#)

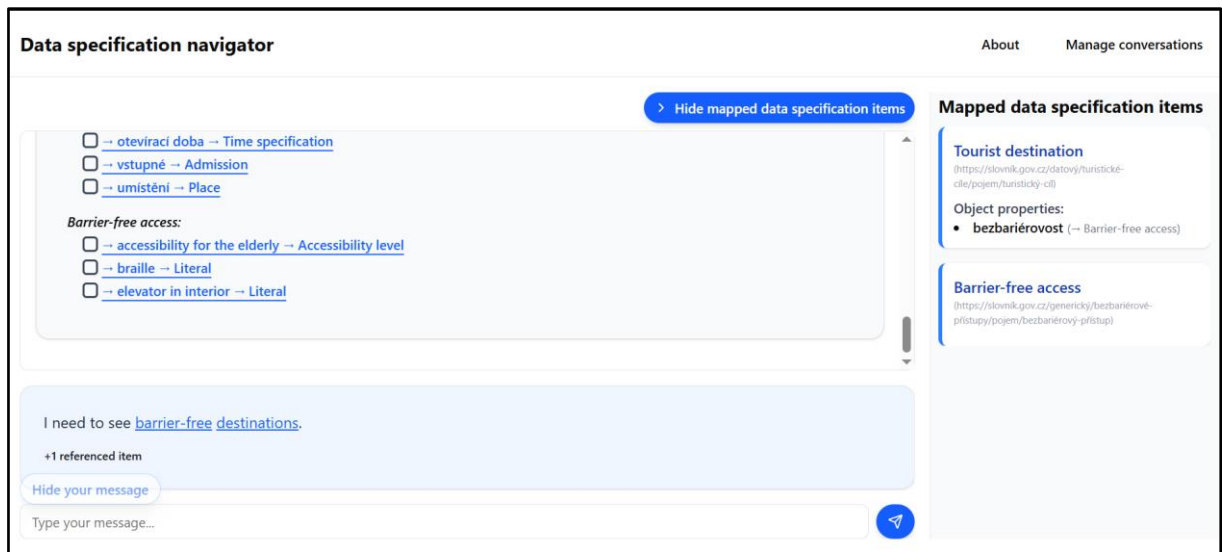
**Barrier-free access:**

- ☐ → [accessibility for the elderly](#) → [Accessibility level](#)
- ☐ → [braille](#) → [Literal](#)
- ☐ → [elevator in interior](#) → [Literal](#)

The "Mapped data specification items" section on the right remains the same as in Figure 6.

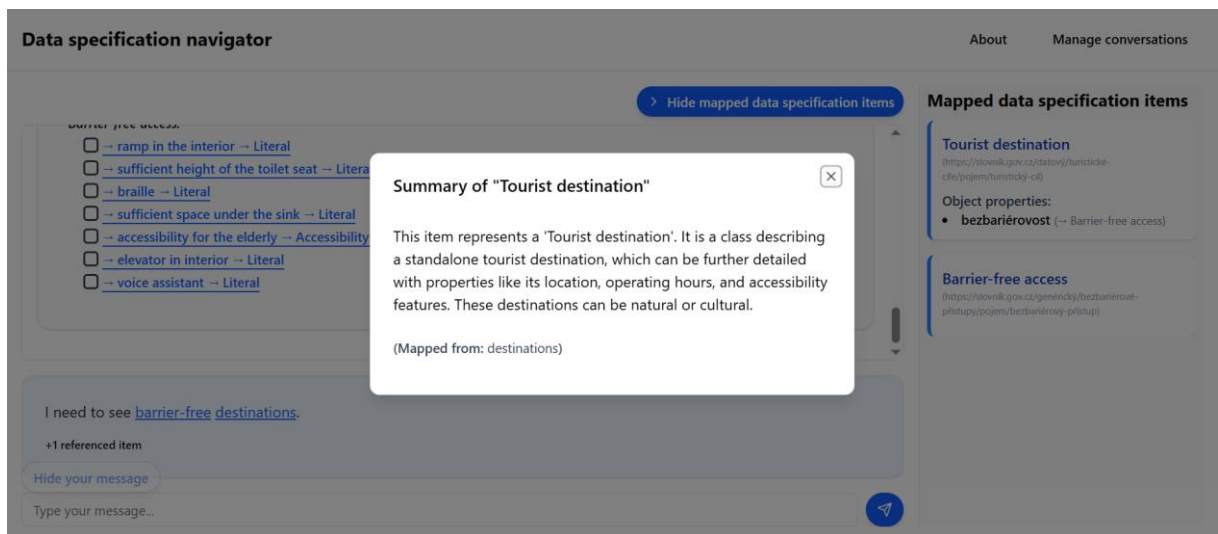
The toggle "Show your message" shows the user message with the highlighted parts corresponding to the mapped items (see Figure 8).

Figure 8: Words highlighting



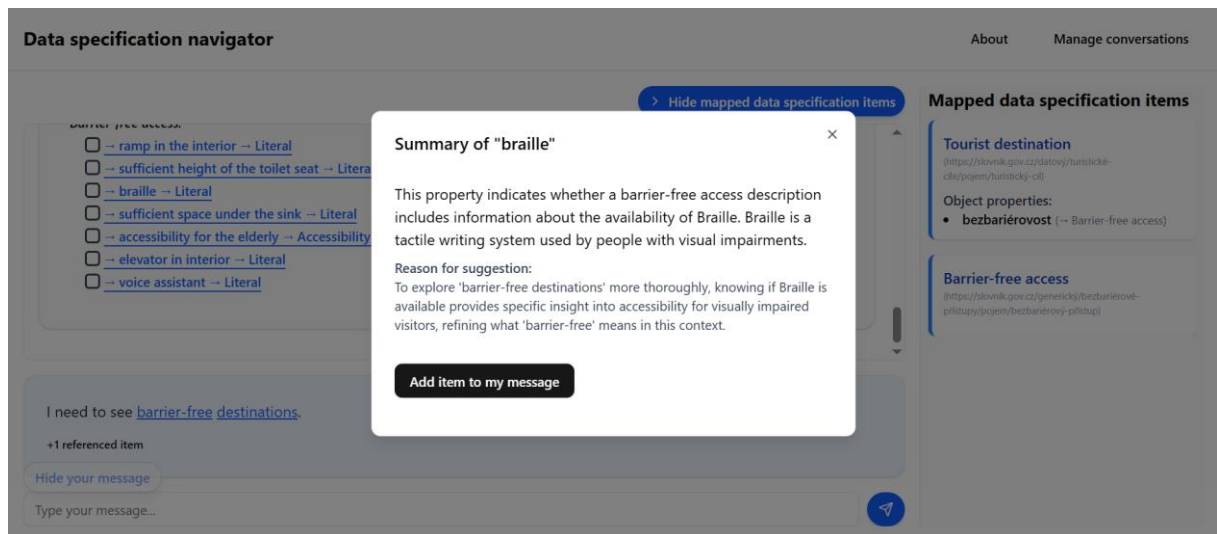
The user can click on a mapped word in their message, and the UI will display a summary for that item (see Figure 9).

Figure 9: Mapped item summary



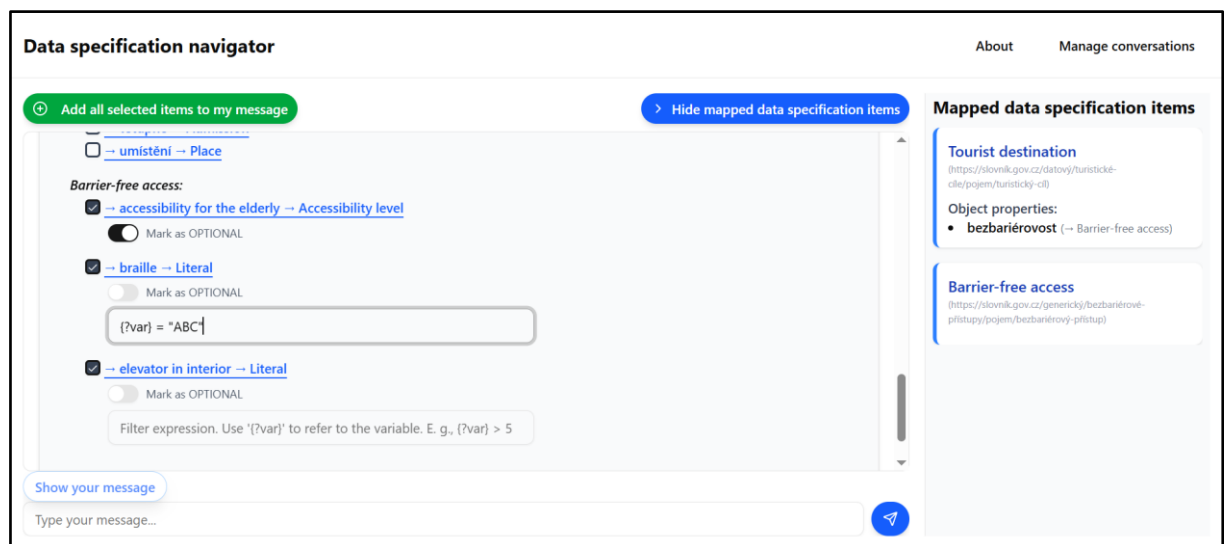
The user can click on a suggestion in the chatbot's reply and the UI will display a summary for that item (see Figure 10).

Figure 10: Suggested item summary



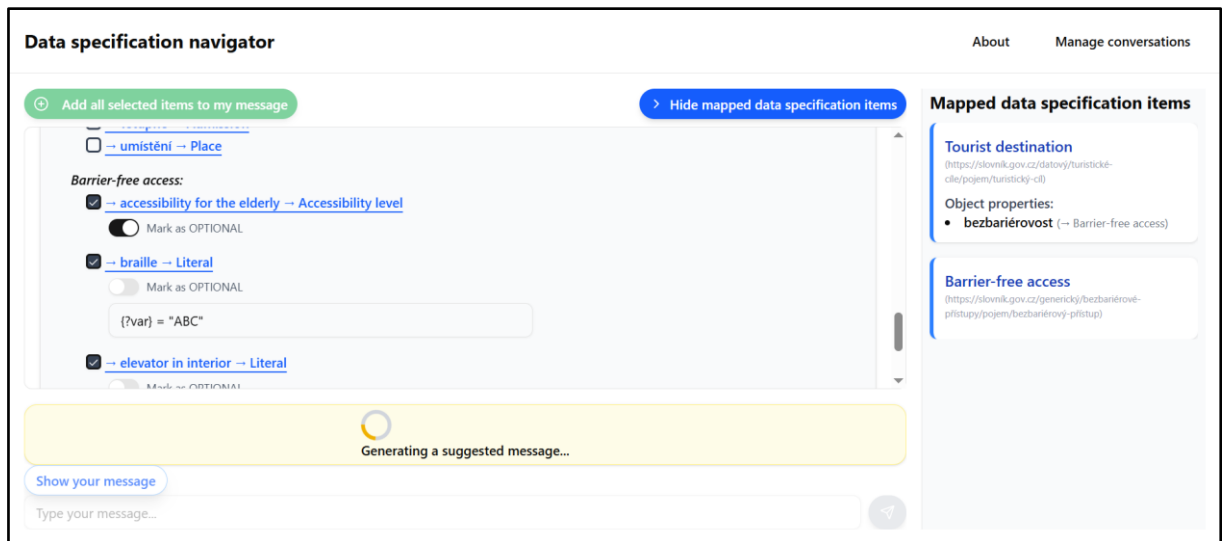
The user selects some suggested items. The UI shows the button “Add all selected items to my message” on top of the chat (see Figure 11).

Figure 11: User chooses suggestions



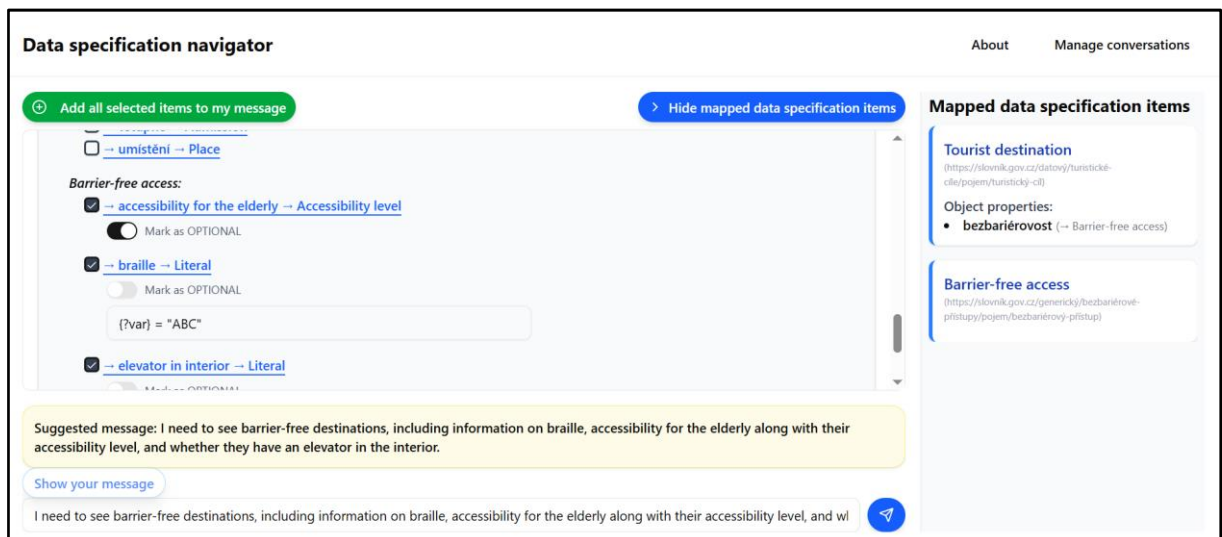
The user clicks the button to add all items, which prompts the UI to retrieve the suggested message from the backend (see Figure 12).

Figure 12: Waiting for a suggested message



The UI displays the received suggested message (see Figure 13).

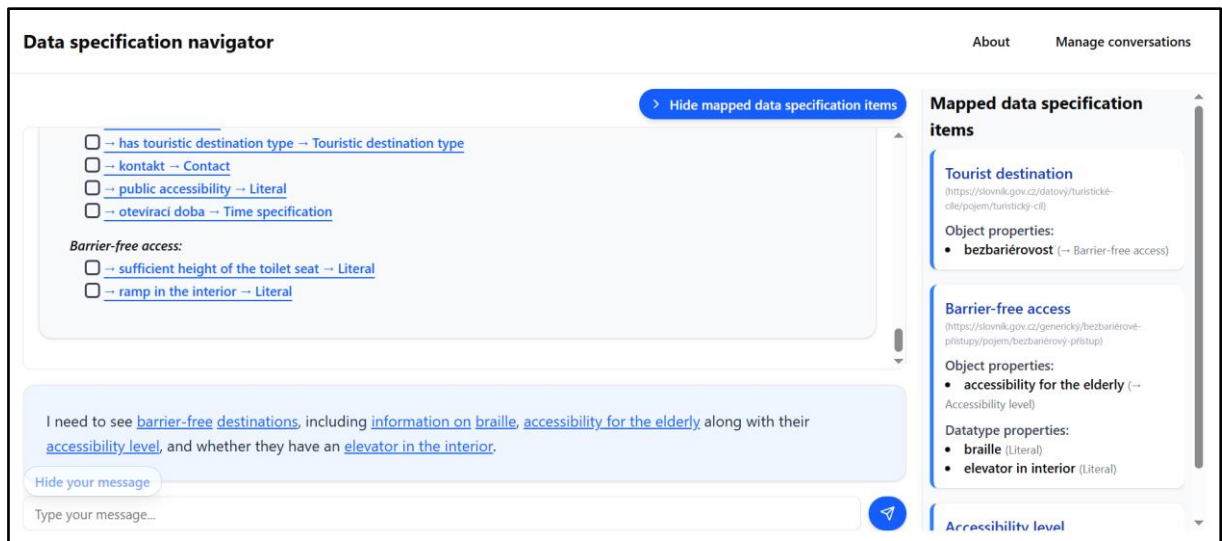
Figure 13: Suggested message displayed



The user sends the suggested message as is. Alternatively, the user can modify the message before sending it.

The chatbot replies with the new answer (see Figure 14). Notice that the right-side panel is updated with newly mapped items.

Figure 14: User sends suggested message and receives a reply



The conversation flow can continue in this manner until the user is satisfied.

### 3 Available operations

This section chapter describes the functionalities of the application and its important UI elements.

#### 3.1 Creating a conversation

To start working with the application, the user needs to navigate to the app's website and create a new conversation. If no conversations have yet been created, the UI will display a short message instructing the user to create a Dataspecer package and then create a new conversation.

To create a new conversation, the user must click on the button "Create new conversation". In the popup box, the user must fill in the correct IRI for the package they want. The field Dataspecer package name does not have to match the actual package's name. The user can choose to fill in anything. This name **does not have any influence** on how the application works nor does it affect any packages in Dataspecer. Once all the fields are filled in, the user can click on the "Create conversation" button. After the conversation has been created, the user can open the conversation to start chatting with the chatbot.

#### 3.2 Conversation

The conversation page is split vertically into two main areas. On the left side are the chat messages and the input box, where the user can type their messages. On the right side is a panel that displays data specification items, which the chatbot has managed to map to from the user messages. This right-side panel can be hidden by clicking on the "Hide mapped data specification items" button.

The first message in the conversation is always sent by the chatbot and has a green background. It informs the user about how many classes and properties that have been mapped in the data specification. Additionally, it gives the user a brief summary of what the data specification is about and suggests a few things the user might want to ask about.



### 3.2.1 Sending a message

The user can type a message into the input box on the bottom of the chatting interface. The user sends their message either by pressing the ENTER key on the keyboard or clicking on the blue button with a paper plane icon next to the input field.

As this application is not intended to be a general purpose chatbot like Gemini or ChatGPT, the user should limit their messages to queries that are relevant to the data specification that they have chosen.

### 3.2.2 Chatbot's reply

If the user sends a message that does not relate to their chosen data specification, the chatbot will reply with a simple answer "I could not find any relevant information in the data specification for your message."

If the chatbot can extract some relevant items from the data specification, its answer will contain the following:

- A SPARQL query (see Figure 6). The user can run this query against a graph database to retrieve data.
- A list of suggestions. These suggestions are the properties from the data specification which will expand the user's current query (see Figure 7). Suggestions are grouped by items that they expand. For example, in Figure 7 the last three suggestions "accessibility for the elderly", "braille" and "elevator in interior" expand the item "Barrier-free access", which was mapped from the user message.

### 3.2.3 Side panel

The right-side panel contains items that have been mapped from the user message. Each card in the panel is a class from the data specification that has been mapped. Each class contains the properties that have been mapped. For example, in Figure 8, the classes "Tourist destination" and "Barrier-free access" were mapped, and the property "bezbariérovost" of class Tourist destination was mapped. When more classes and properties are mapped, the UI will add them to this side panel.

### 3.2.4 User message highlighting

When user sends a message, it gets added to the conversation history, as can be seen in Figure 5. When the chatbot sends an answer, the reply message is long, and the user message is scrolled out of view. In this case, the UI displays the most recent user message directly above the text input field for user messages (see Figure 8). This message is displayed with certain words highlighted. These are the words that the chatbot has mapped to items from the data specification. The user can click on each highlighted word to display a summary.

If an item is referenced in the user message but doesn't have a direct word-for-word mapping, it may be listed under the "+1 referenced item" button below the message text.

### 3.2.5 Suggested items

The user can add suggestions from the chatbot to their message, thus expanding it. The user selects suggestions by ticking the checkbox next to the suggestion they want to add. For each suggestion, the user can specify whether to add this item as OPTIONAL, which results in the item being in an OPTIONAL branch in the generated SPARQL query (see Figure 11).

When viewing a summary for a suggested item, the user can click on the “Add item to my message” button in the summary window (see Figure 10). This action is equivalent to ticking the checkbox next to the suggested item in the chatbot’s reply.

#### 3.2.5.1 Filtering datatype properties

If the user selects an item that is a datatype property (like a number or text), they have the option to add a filter, which helps narrow down the query. Use **{?var}** to refer to the property being filtered.

Example: Let's say the user wants to find people named “John”.

1. Select the property “name” from suggested items.
2. The text box for filtering will appear below the suggestion.
3. In this text box, type **{?var} = “John”**.

#### 3.2.5.2 Confirming selections

The user confirms all their selected items by clicking on the green “Add all selected items to my message”. This does not send a new message to the conversation. This action will prompt the chatbot to generate a suggested message, which is displayed to the user above the text input field. This suggested message is the user’s most recent message expanded by the items they have selected. If the user is happy with that message, they can send it to the chatbot without any modifications.

## 4 Best practices

While you can edit the suggested message before you send it, this is generally not recommended.

The chatbot is designed to build on its own suggestions, which makes it much faster and more accurate. When you make significant changes to the suggested message, the chatbot must re-evaluate everything from the beginning, which can slow down the response time and may lead to less accurate results.

## 5 Limitations

This chapter discusses the performance bottlenecks and unsupported scenarios in the application.

### 5.1 Performance

Behind the scenes the chatbot works with a large language model (commonly referred to as AI) to map messages to the data specification items. Depending on the AI being used, the chatbot might take a long time to answer each user message.

### 5.2 Limited query support

The system is unable to generate correct SPARQL queries for certain cases. For example, a query about a person who is a “boss of another person” would incorrectly translate into a query about a person who is their own boss.