

## **EE3206 Project Report**

### **A Java Program for AI Chat History Management**

#### **Group Members:**

**SHEN Yichen 57124265**

**ZHANG Jiachang 57124880**

**(Equal contributions)**

#### **Supervisor:**

**Dr. LI Haoliang**

## I. Introduction

With the general availability of conversational AI, there is one more way that we can get information, via chats with such AIs. And currently, there are hundreds of, if not thousands of, platforms that you can chat with such a bot. But from our observation, different platforms have various and uneven ways to store your chat. Like in the simplest form, there is no place to login, and these dialogues are just gone if you start a new chat. For some other platforms, they may only store the most recent 20 or 40 dialogues in total. And even with these platforms able to keep track of all your history data, a search function may not be implemented. And in some cases, if you can make a search through your chat histories, there may be no need to make a chat completion request again or you can start with more detailed context discussed with the bot in the past. And this can also help the model providers to save the cost of hosting LLMs. And a further issue is that it is almost impossible to get all the data across so many platforms all at once. Although people may argue that they can use word or other means to move the data to their own storage, it still takes more effort and may not be convenient to search all these history data across multiple chat records. So we came up with this AI dialogue tracker as our first attempt to better organize chat information with AI and ultimately increase productivity in chats with AI.

And from a high level point of view, this tracker can search through the dialogue and return matching dialogues matching the search queries, which is implemented as the main window. Users can select a dialogue to open and see the detailed content in a new window. There is also an UI for adding on a new dialogue by clicking the “add dialogue” button. And in the backend, to manage the chat records and make them searchable, a json approach is used to keep multiple rounds of a dialogue. And even if the program shuts down by exiting, the json file is still there so that users can view chat next time they launch the program.

## II. Implementation details

### *a) Introduction to classes*

The fundamental unit of our program is the Dialog class, which acts as both the storage and manipulation unit. Each Dialog consists of a title and a list of Conversation objects, where each Conversation captures a single round of question and answer. The structure is as follows:

- Dialog
  - Title
  - List<Conversation>
- Conversaion

- Question
- Answer

The FileManager class serves as the data manipulation hub. It provides a variety of methods for reading, writing, and adding conversations to dialog files. Additionally, it includes a search method that acts as the engine for the core functionality of the program—searching through dialogs, on everything (both dialog title and contents).

We have designed three classes, each responsible for a specific GUI window:

1. **MainWindow:** This is the startup window of our program, providing users access to search dialogs. It extends JFrame and incorporates elements such as buttons and text fields, implemented through the members of the JFrame class. Key methods include:

- **setupUI:** This method arranges all elements and attaches event listeners.
- **searchDialogs:** It utilizes methods from FileManager to search and display results.
- Additional methods facilitate a smooth user interaction experience and navigation to other windows.

And there are some other peripheral methods to realize smooth user interactive experience and jumping to other windows.

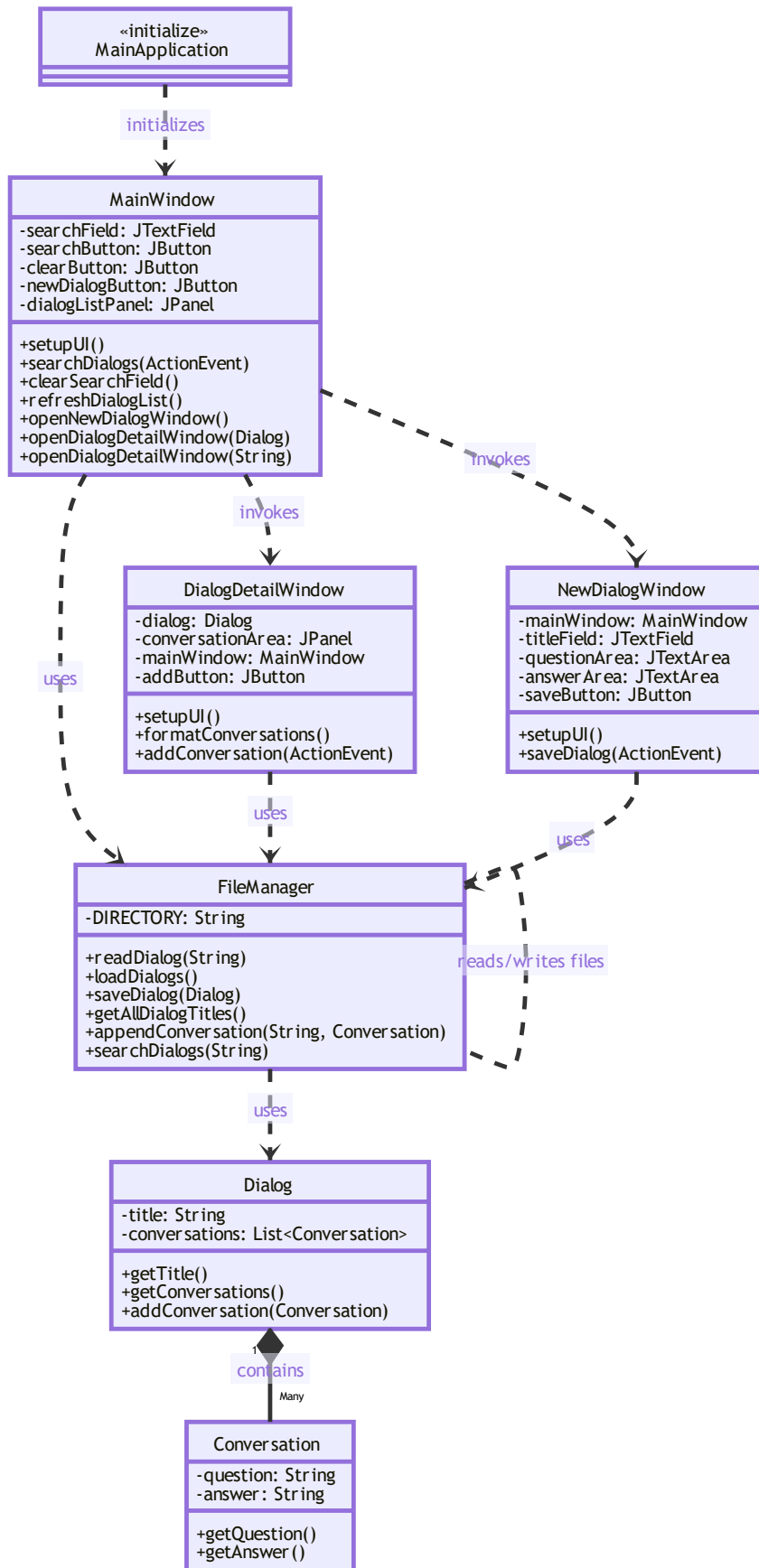
2. **NewDialogWindow:** Accessed from the MainWindow via a dedicated button, this window allows users to input new dialogs. Its design mirrors that of the MainWindow, extending JFrame and including methods essential for adding new dialog entries.
3. **DialogDetailWindow:** Also accessible from the MainWindow, this window displays an existing dialog, offering users the option to add more rounds of conversations. Its structure is consistent with the other GUI windows.

#### *b) File management and exception control*

To ensure that dialog data is managed in a clear and easily maintainable manner, we use JSON files for storage. As a widely accepted data format, JSON not only enhances clarity but also ensures broad applicability. All the dialog JSON files are stored in “dialog” directory under the working directory.

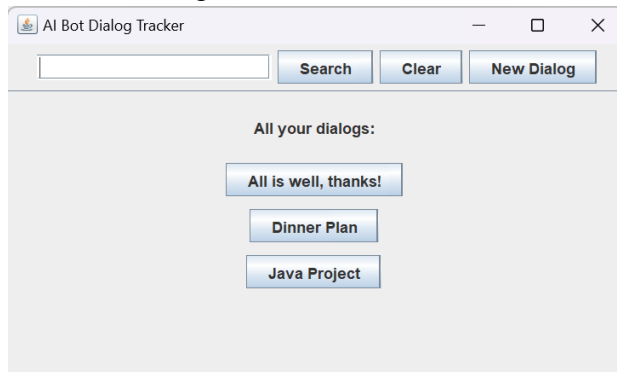
We have implemented robust exception handling to address potential user errors, particularly regarding empty inputs in text fields. These controls ensure that the program remains operational and does not crash during unexpected user interactions.

### **III. CRC Cards and UML**

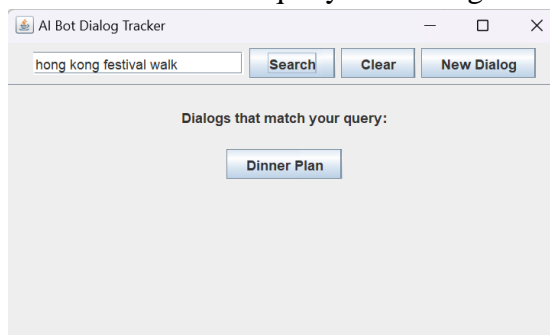


## IV. Functions Demonstration

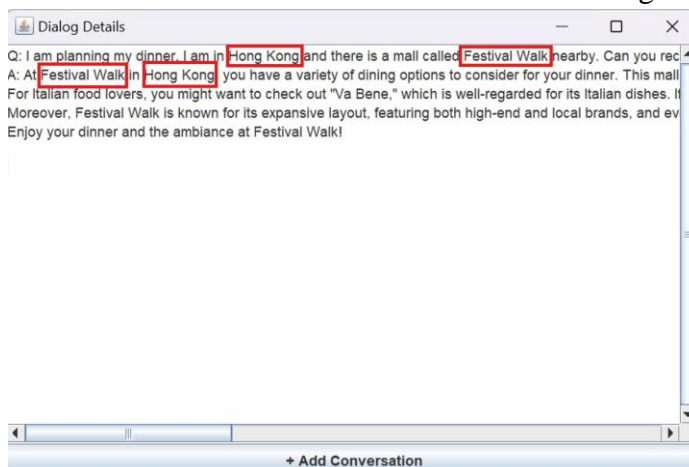
After launching the program, this will be guided to the start page with search and you can view dialogs with titles.



And by typing the search keyword “hong kong festival walk”, it will search and find out a dialog with word “Hong Kong” and also “Festival Walk” (case insensitive and sub-word match for query and dialogue content):



And click the button to see the detail of this dialogue:



Click “+ Add Conversation” to add one more round of dialogue with the AI, which you need to add your question and then answer by the AI:

Dialog Details

Q: I am planning my dinner. I am in Hong Kong and there is a mall called Festival Walk nearby. Can you recommend some good food there?  
A: At Festival Walk in Hong Kong, you have a variety of dining options to consider for your dinner. This mall features over 27 restaurants offering a variety of cuisines. For Italian food lovers, you might want to check out "Via Bene," which is well-regarded for its Italian dishes. If you're in the mood for Japanese cuisine, you might want to check out "Sakura," which is known for its authentic Japanese dishes. Moreover, Festival Walk is known for its expansive layout, featuring both high-end and local brands, and even includes an ice skating rink and a cinema. Enjoy your dinner and the ambiance at Festival Walk!

输入

Enter your question:  
in Kowloon Tong to have a dinner?

确定 取消

+ Add Conversation

Dialog Details

Q: I am planning my dinner. I am in Hong Kong and there is a mall called Festival Walk nearby. Can you recommend some good food there?  
A: At Festival Walk in Hong Kong, you have a variety of dining options to consider for your dinner. This mall features over 27 restaurants offering a variety of cuisines. For Italian food lovers, you might want to check out "Via Bene," which is well-regarded for its Italian dishes. If you're in the mood for Japanese cuisine, you might want to check out "Sakura," which is known for its authentic Japanese dishes. Moreover, Festival Walk is known for its expansive layout, featuring both high-end and local brands, and even includes an ice skating rink and a cinema. Enjoy your dinner and the ambiance at Festival Walk!

输入

Enter AI's answer:  
multiple dining opportunities for visitors

确定 取消

+ Add Conversation

And details after adding the new round of dialogue:

Dialog Details

Q: I am planning my dinner. I am in Hong Kong and there is a mall called Festival Walk nearby. Can you recommend some good food there?  
A: At Festival Walk in Hong Kong, you have a variety of dining options to consider for your dinner. This mall features over 27 restaurants offering a variety of cuisines. For Italian food lovers, you might want to check out "Via Bene," which is well-regarded for its Italian dishes. If you're in the mood for Japanese cuisine, you might want to check out "Sakura," which is known for its authentic Japanese dishes. Moreover, Festival Walk is known for its expansive layout, featuring both high-end and local brands, and even includes an ice skating rink and a cinema. Enjoy your dinner and the ambiance at Festival Walk!

Q: Besides Festival Walk Hong Kong, are there some other malls nearby in Kowloon Tong to have a dinner?  
A: Based on the search results, there are a few other notable shopping malls located near Festival Walk in the Kowloon Tong area that offer dining options.

+ Add Conversation

After adding, you can exit by the close window, and then press the clear button to go back to the starter page.

Now suppose there is a new round of chat that you made with AI, you can press New dialog to create a record for new dialogues by writing title, a question-answer pair, and to click save to write a new json file in the file manager.

The 'New Dialog' window has three main sections: 'Title', 'Question', and 'Answer'. The 'Title' field contains the text 'Solar system meaning'. The 'Question' field contains the text 'What is a solar system?'. The 'Answer' field contains the text 'A solar system is a collection of objects that orbit a central star, such as the Sun in our solar system. It typic'. A 'Save' button is located at the bottom right of the window.

And please be noticed that when creating record for a dialogue, you need to fill all these three fields, otherwise it will raise an warning window:

The 'New Dialog' window is shown with an error message overlay. The error message box has a red 'X' icon and the text 'Please fill all fields'. The 'Title' field is empty. The 'Question' and 'Answer' fields are also empty. A 'Save' button is visible at the bottom right of the window.

After completing the creation, it will be reflected on the starter page immediately:





The 'AI Bot Dialog Tracker' window displays a list of dialog records under the heading 'All your dialogs:'. The records are: 'All is well, thanks!', 'Dinner Plan', 'Java Project', and 'Solar system meaning'. Each record is represented by a button. At the top of the window, there is a search bar and three buttons: 'Search', 'Clear', and 'New Dialog'.

Moreover, at the dialog file, you can see the corresponding json file to store the chat history data.

Before execution:

 All is well, thanks!.json	2024/4/21 20:13
 Dinner Plan.json	2024/4/21 21:14
 Java Project.json	2024/4/21 20:13

After execution:

 All is well, thanks!.json	2024/4/21 20:13
 Dinner Plan.json	2024/4/21 21:14
 Java Project.json	2024/4/21 20:13
 Solar system meaning.json	2024/4/21 21:37

## V. Future improvements

### a) *UI/UX optimizations*

Our program just implements a simple UI, and we know that it is far from optimal. There is huge space for improvement.

In the starter page, the button for displaying different dialogue titles could be set as the same width, and the ordering of the dialogue can be rearranged according to different ordering, like time, alphabet order, length of dialogue etc.

And in the page for displaying detailed dialogue, the input box is too wide and you need to scroll to the right to see all content of a dialogue. Also there is no clear separation between the words by user and the words by AI. We may consider advanced UI to separate them.

In the adding dialogue window, it has the same problem just as the display detailed chat window, and here after viewing this, the two windows could be improved by combining into one window by considering a more intuitive design to add dialogue no matter if the dialogue is from scratch or is being edited.

Also, there is no support for markdown, so this UI just will display plain text, which is not the case for display in most platforms, however, implementation of this requires a far more advanced framework than JWT and Swing.

From the perspective of UX, sometimes the user may want to include more information, like some platforms supporting chat link sharing, like model details and high level description for the dialogue, and this could be done by adding some extra attributes.

### b) *Optimization on search functions*

The backend file manager in our project is just to search for exact match or sub-word exact match in the dialogues. And this search method is implemented by iterating through all the words in all the files. The scale may be limited to only several KBs. A potential optimization is to use a multi-threading approach, which may be useful when there are several MBs of dialogue stored. And if the scale goes further, some techniques from database systems may be applied, and this is far away from our scope.



And it is also noticed that sometimes the user may not enter exact match keywords, and sometime will enter some less important keywords like “this”, “that”, “the”, and such keyword and similar word match search can be implemented by some NLP related functions, from TF-IDF to filter out unimportant words, to vector embedding retrieval to get the most relevant terms.

c) *Further improvements*

Besides just tracking the data from chat records, one may obviously consider designing a function to continue chatting with an AI model, but this will require using other packages and SDKs, and will introduce a lot of external open source code. However, such a suggestion is really needed beyond just for completion of this project.

It is also noticed that for some chatbot platforms, the bot may accept multimedia inputs, and a completely new framework may be needed to display these kinds of multimedia. And in some UI designs for chatbots, there is a hierarchy of folders to store dialogue files, or for a few designs, one question may be asked and retried generating outputs several times, and under one answer, the user may ask several questions under the same reference context. These two features can be implemented by modifying the json file format, and introducing a hierarchical structure in folders, but the optimization for search may also be changed due to the format and other stuff.