## Architectural Overview

The project consists of several key components, each interacting to deliver a seamless user experience. Here's a breakdown of the components and their interactions:

**Components:**

1. **User Interface (UI)**

* A user-friendly chat application interface.
* Allows users to input queries and receive responses.
* Displays the chat history.

1. **Chat Bot**

* Main engine for handling user interactions.
* Uses natural language processing (NLP) to understand user queries.
* Routes queries to the appropriate backend services.

1. **Knowledge Base (ChromaDB)**

* Stores documents with data on bus schedules, routes, fares, etc.
* Ensures efficient storage and retrieval of information.

1. **Query Handler**

* Processes user queries.
* First attempts to retrieve information from the Knowledge Base.
* If information is not found, makes external API calls.
* If the API does not provide the required information, performs a search on DuckDuckGo.

1. **External APIs**

* Sources of additional bus-related information not available in the Knowledge Base.
* Used when the Knowledge Base does not have the requested information.

1. **DuckDuckGo Search Integration**

* Provides a fallback mechanism to search for information on the web.
* Used if both the Knowledge Base and external APIs do not provide satisfactory responses.

1. **Chat History Database**

* Maintains a log of all interactions with timestamps and user inputs.
* Allows for future reference and analysis.

**Workflow:**

1. **User Query Input:**

* The user inputs a query through the chat application interface.

1. **Query Processing:**

* The Chat Bot uses NLP to understand the query and forwards it to the Query Handler.

1. **Knowledge Base Retrieval:**

* The Query Handler first attempts to retrieve the requested information from the Knowledge Base (ChromaDB).
* If the information is found, it is presented to the user.

1. **External API Call:**

* If the information is not found in the Knowledge Base, the Query Handler makes an external API call to fetch the required data.
* If the API provides the information, it is presented to the user.

1. **DuckDuckGo Search:**

* If the external API does not provide satisfactory information, the Query Handler performs a search on DuckDuckGo.
* The results are processed and the relevant information is presented to the user.

1. **Response to User:**

* The Chat Bot compiles the response and sends it back to the user through the chat application interface.

1. **Chat History Logging:**

* The conversation is logged in the Chat History Database with timestamps and user inputs for future reference.

**Flowchart**

Below is a simplified flowchart depicting the system's workflow:

+------------------+

| User Interface |

+------------------+

|

v

+------------------+

| Chat Bot |

+------------------+

|

v

+------------------+

| Query Handler |

+------------------+

|

v

+-------------------------+---------------------------+

| | |

v v v

+------------------+ +---------------------+ +-------------------+

| Knowledge Base | | External APIs | | DuckDuckGo Search |

| (ChromeDB) | | | | |

+------------------+ +---------------------+ +-------------------+

\ | /

\ | /

\ | /

\ | /

\ | /

\ | /

v v v

+------------------------+

| Response Compilation |

+------------------------+

|

v

+------------------------+

| Response to User |

+------------------------+

|

v

+------------------------+

| Chat History Logging |

+------------------------+