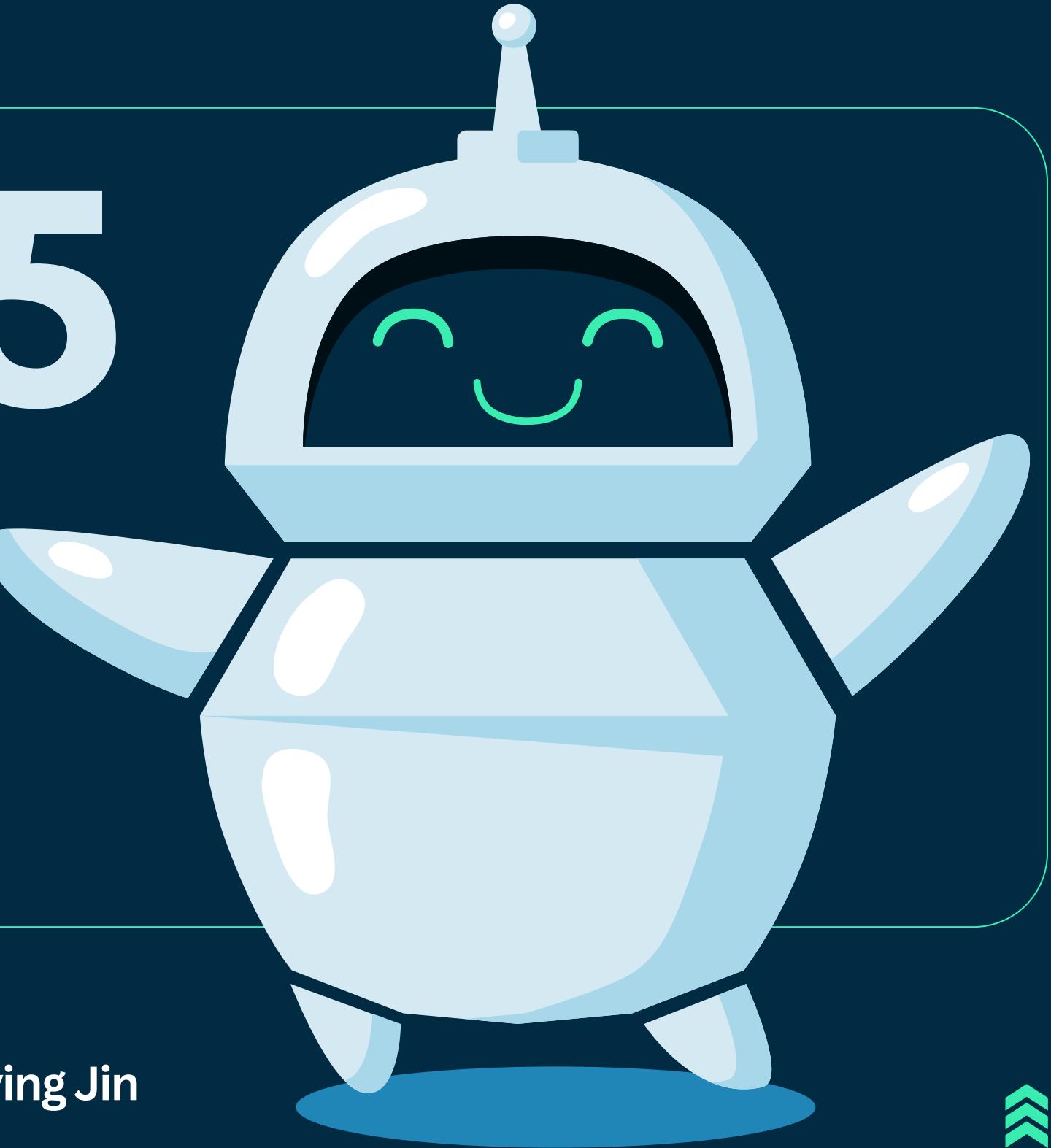


INFO 6205

BUGFREE

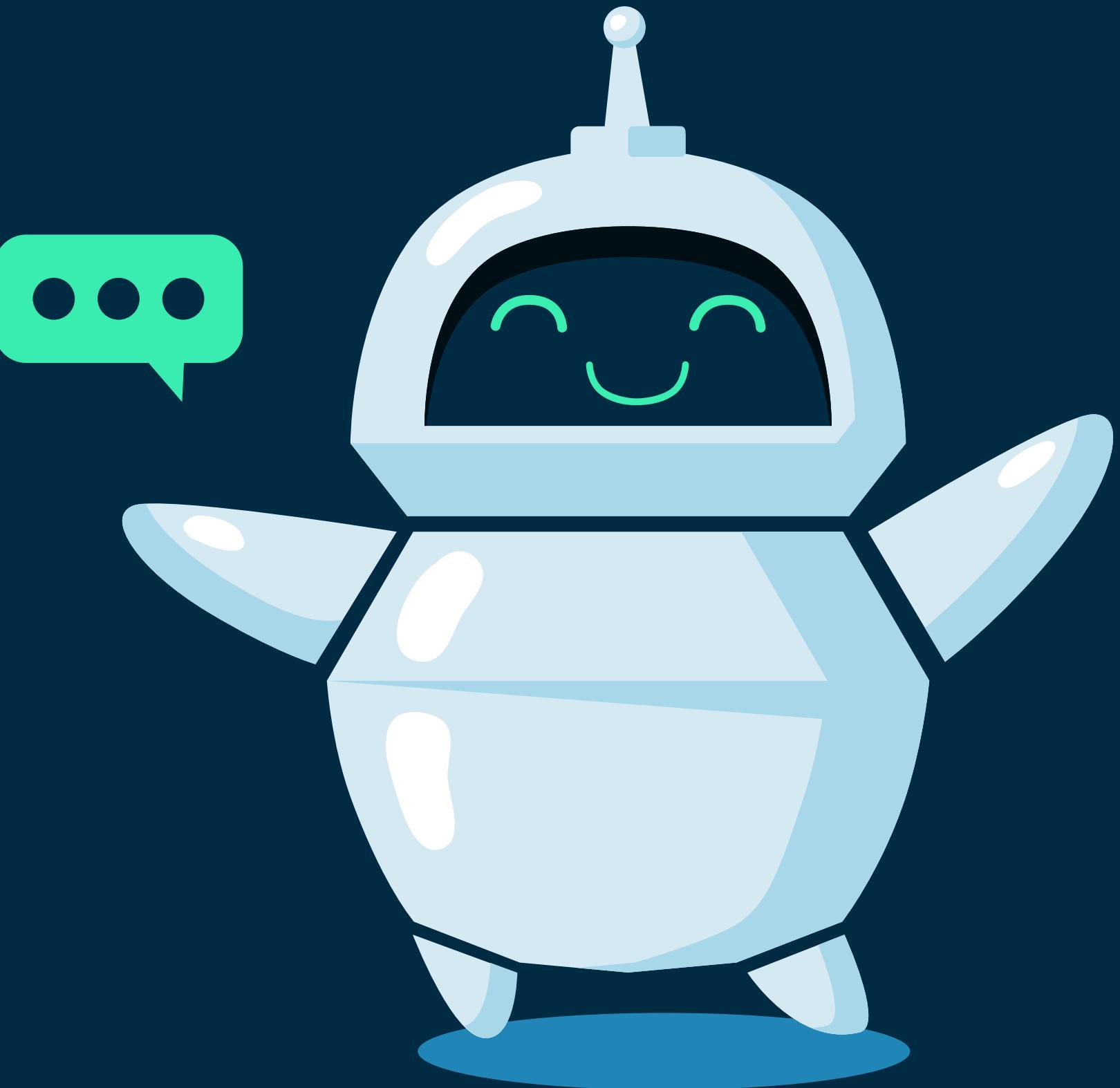
AI Chatbot solution

Jiaye Lee, Qiuyue Zhong, Yuying Jin



OUTLINE

1. Problem Description
2. Analysis
3. Design
4. Implementation
5. Results
6. Discussion
7. Conclusions
8. Job assignments



INTRODUCTION



Users in various field of systems need to retrieve specific pieces of information accurately and quickly.



Traditional search systems and static databases can be limited in their ability to provide context-aware and natural language responses.



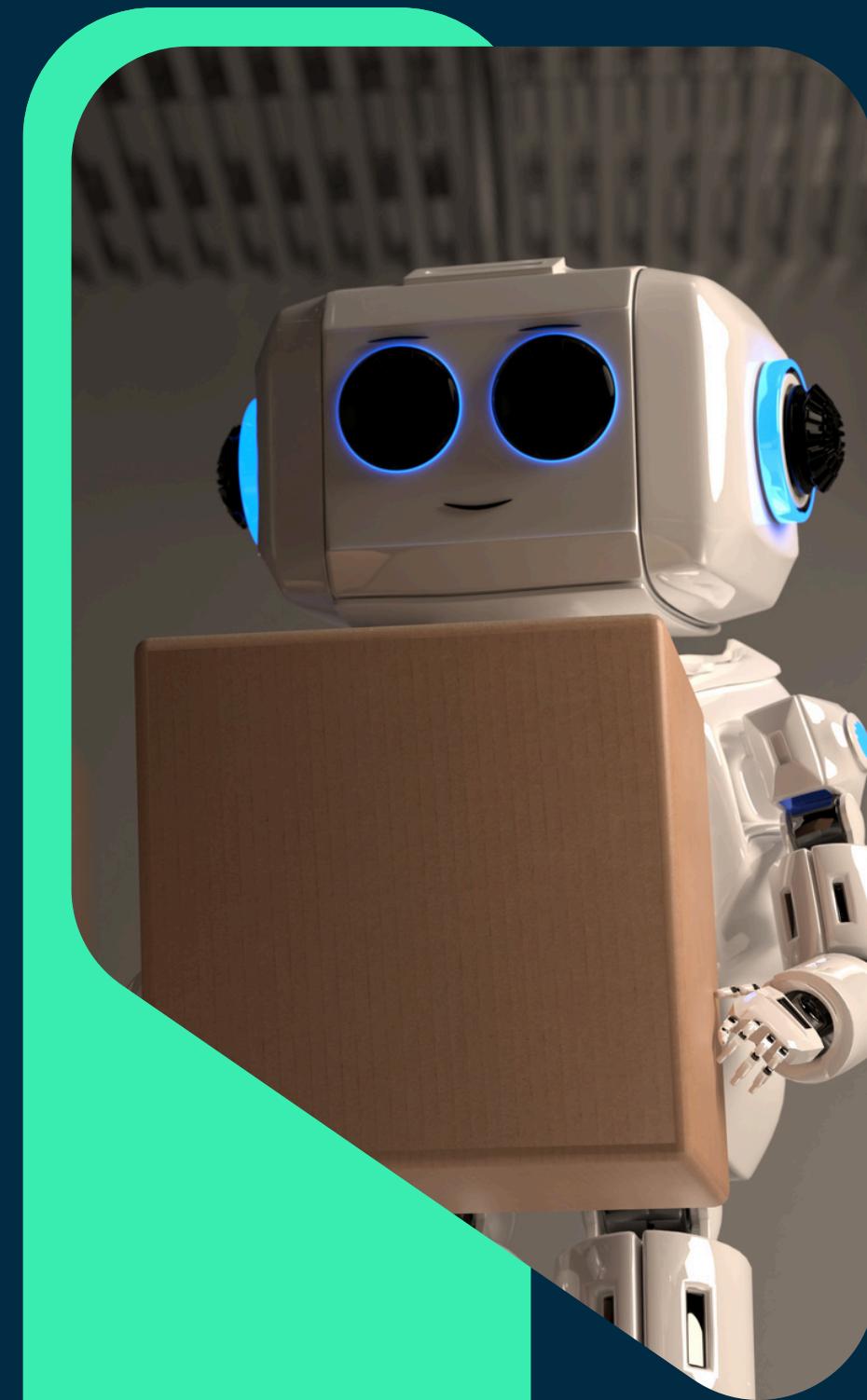
An **AI chatbot** powered by LLMs enables domain-specific knowledge retrieval and natural language interaction.

ANALYSIS



- Difference of Queue/PriorityQueue
- Merge Sort
- Dynamic Knowledge Base
- Question & Answering System
- Why arraylist not linkedlist

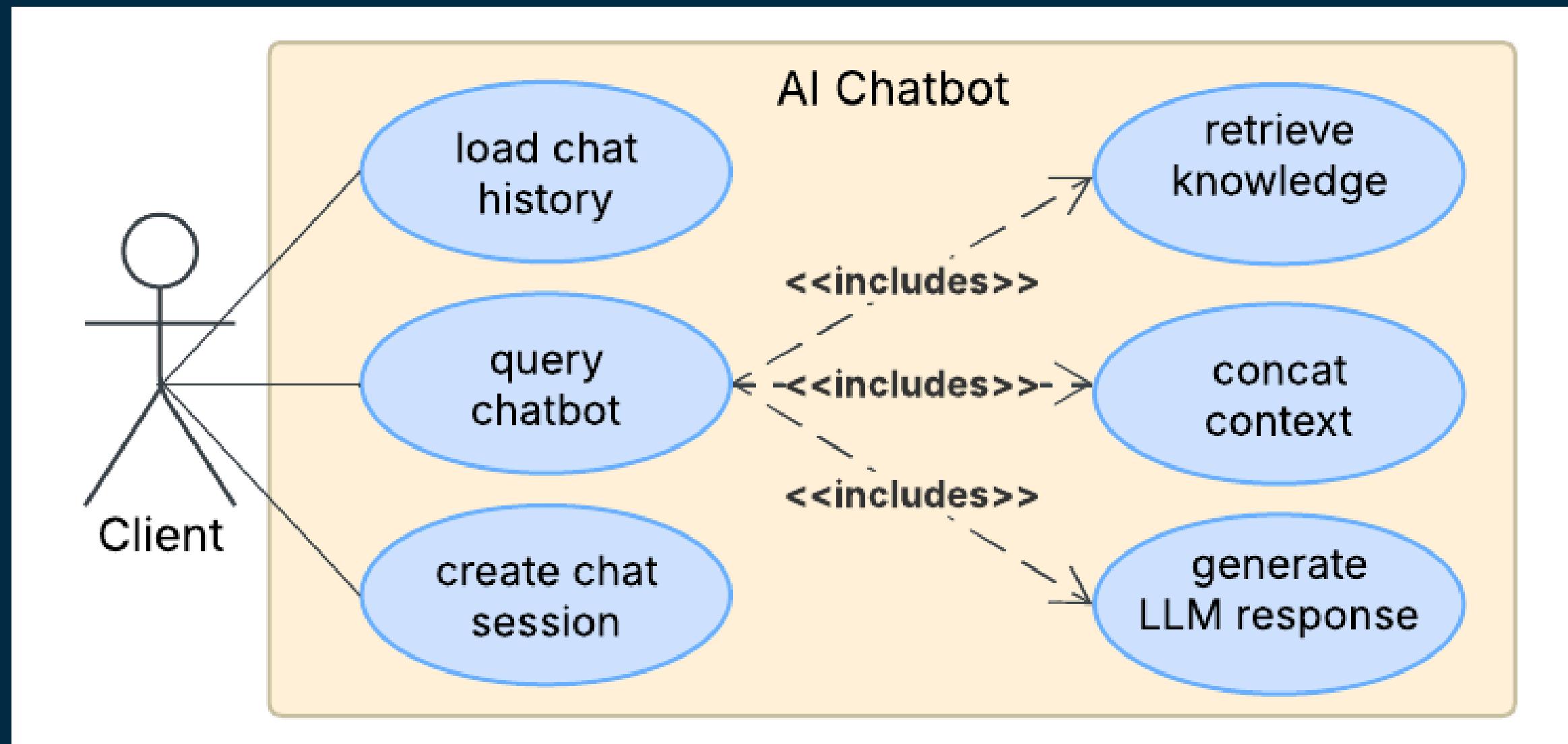
DESIGN



Covered topic:

1. Eclipse
2. GUI
3. Recursion
4. Queue
5. Sorting
6. List
7. Hashing

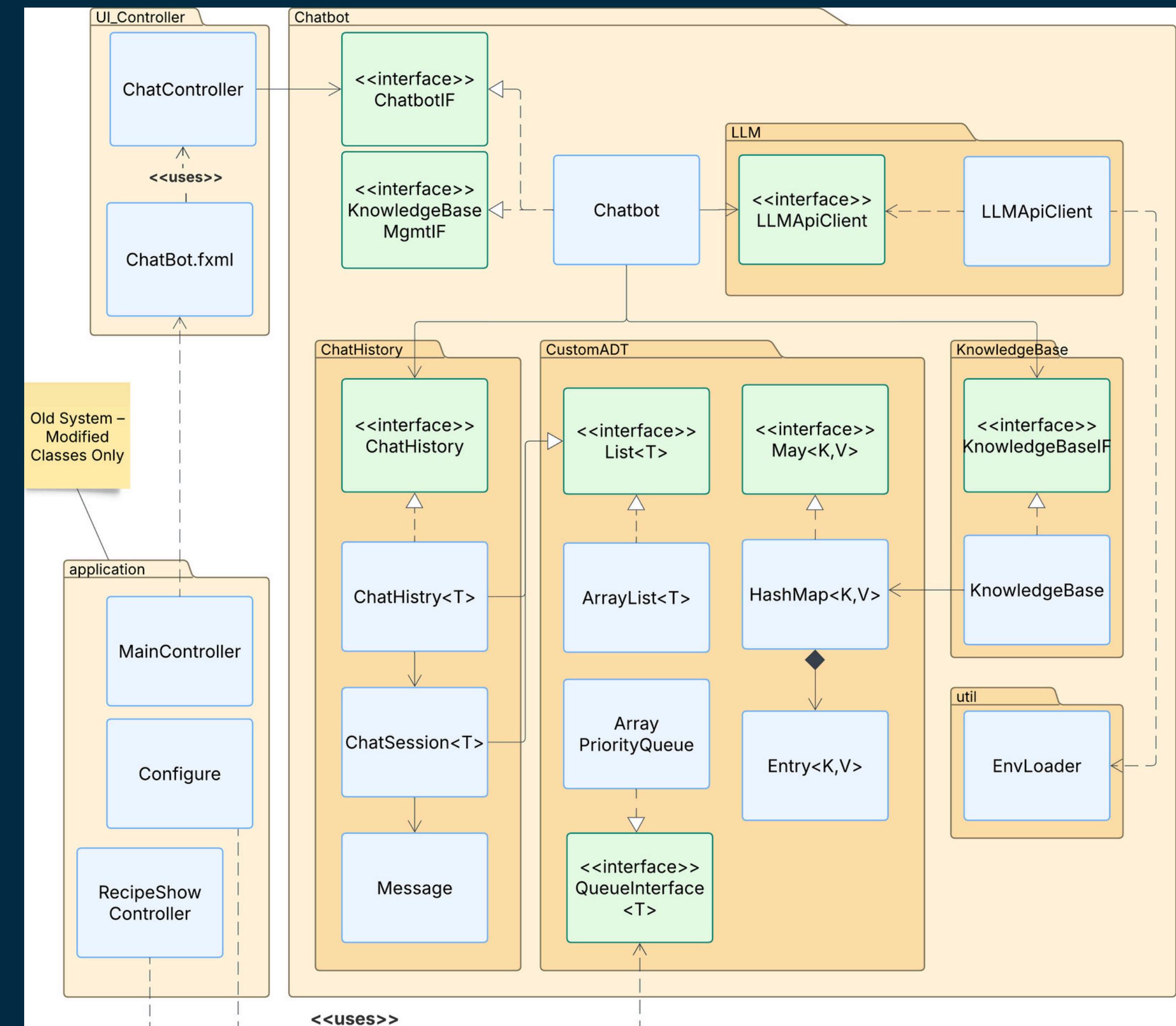
USE CASE



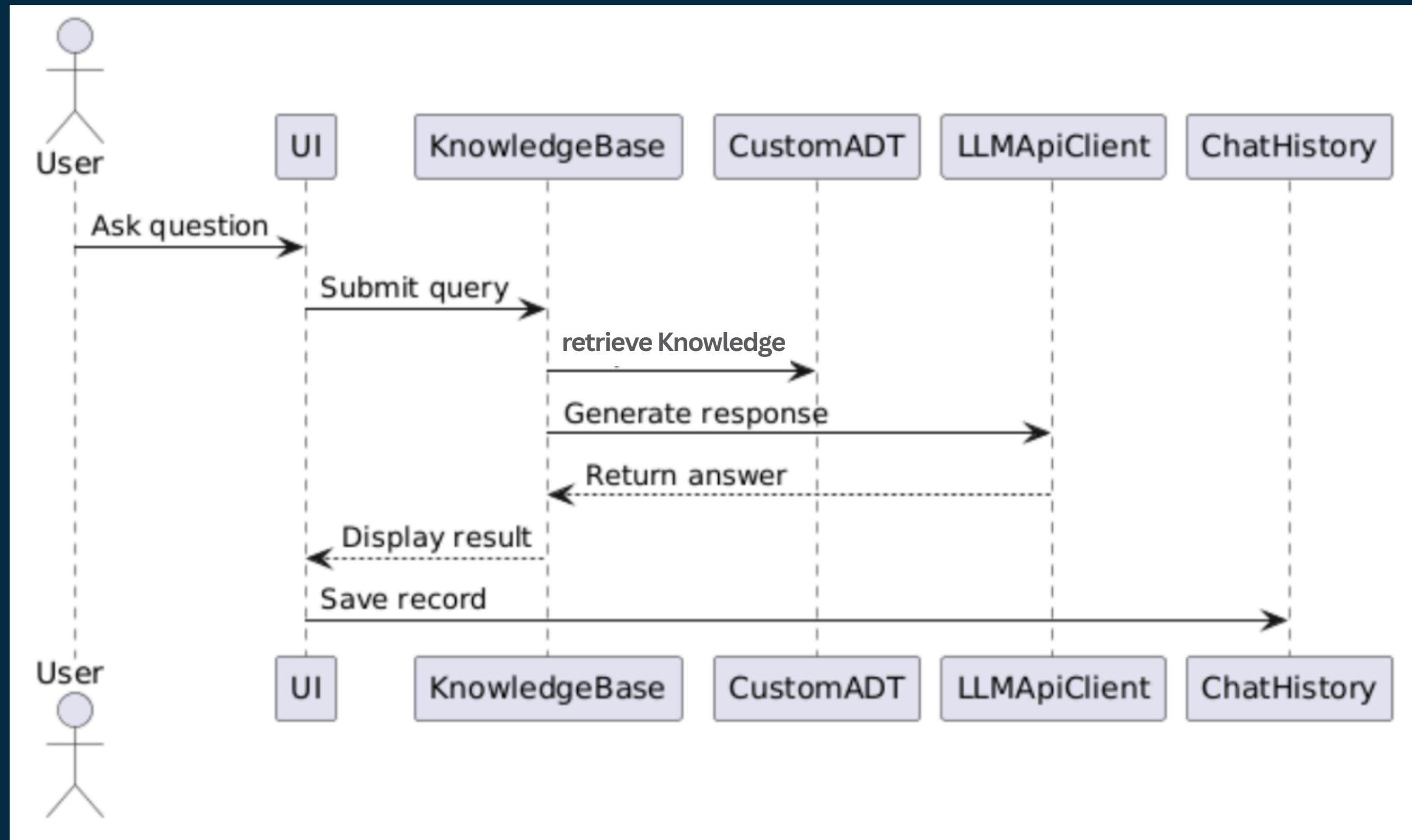
SYSTEM DESIGN

Main Modules

- UI Controller
- LLMApiClient
- ChatHistory
- Knowledge Base
- CustomADT



Q&A WORKFLOW



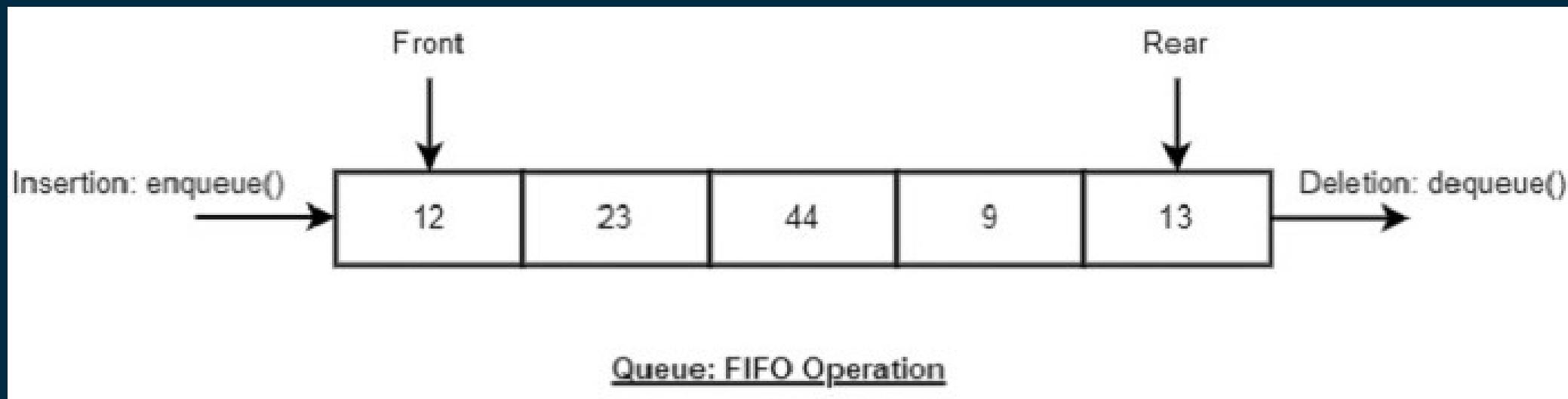
QUEUE

What is Queue

A Queue is a linear data structure that follows the FIFO (First-In, First-Out) principle.

In a Queue:

- Enqueue -> Add an element at the rear.
- Dequeue -> Remove an element from the front.



Why Queue?

1. Ordering Recipes by Popularity
2. Efficient Resource Management
3. Handling Dynamic Data

Key operations of Queue

- isEmpty()
- add(T element)
- T remove()
- displayQueue()
- getTopThreeRecipes()

MERGE SORT & RECURSION

What is MergeSort

It is a divide-and-conquer algorithm that splits the array into halves, sorts each half, and then merges them back together in order.

How we use it in the project

It sorts our Recipe objects by their favorite count, from highest to lowest.

Why MergeSort?

- Efficient for sorting large data sets
- Consistent performance: $O(n \log n)$ time even in the worst case

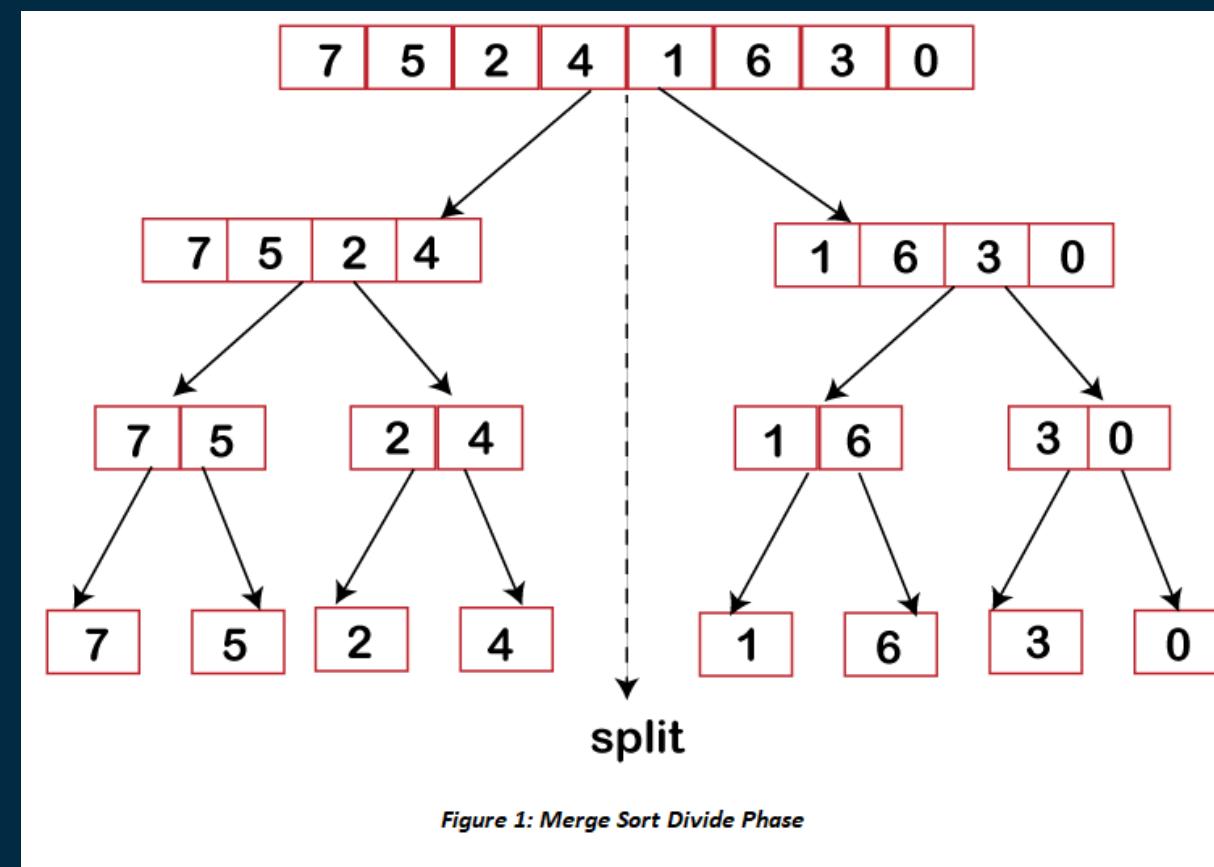


Figure 1: Merge Sort Divide Phase

LIST & ARRAYLIST

What

`ArrayList<T>` is a dynamic array implementation of `List<T>`, using an underlying Object array that resizes automatically

Why

1. Consistent Data Handling across sessions and messages
2. Customizable Behavior with dynamic resizing and type safety
3. Clean OOP Design via interface + implementation separation
4. Code Reuse in multiple classes (`ChatHistory`, `ChatSession`)

⚙️ Key Methods of ArrayList

- `add(T entry)`
- `get(int index)`
- `remove(int index)`
- `remove(T entry)`
- `contains(T entry)`
- `contains(T entry)`
- `toArray()`
- `indexOf(T entry)`
- `clear()`
- `size() / isEmpty()`



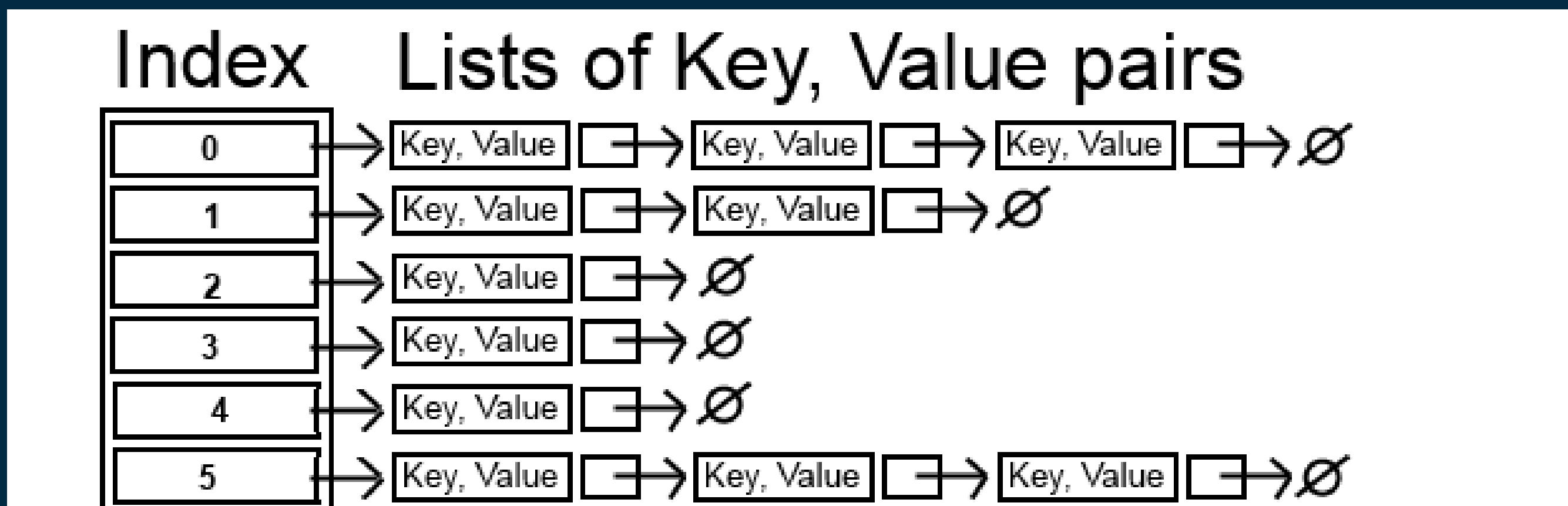
HASHING

What is Hashing?

- Map keys to indices via hash function
- Enables fast $O(1)$ access in average case

Why Hashing?

- Efficient lookup and update
- Scales with growing data
- Ideal for knowledge storage



HASH MAP

Implementation:

Hash Map with ArrayList

Main operations

- put(key, value)
- get(key)
- remove(key)
- isEmpty()
- keys()
- values()
- hash(key)
- resize()

How Hashing Powers Our Knowledge Base

- Entry as Key: Object (e.g., Recipe) mapped to a unique key.
- String Value: Attributes (e.g., ingredients, directions) encoded as strings.
- Hashing: The hash function maps keys to locations in internal buckets (ArrayList).
- Key operations of knowledge base:
 - addEntry(): insert encoded object
 - updateEntry(): overwrite value
 - retrieveAll(): return all key-value pairs
 - searchByInfo(): search values by substring

IMPLEMENTATION

API

Gemini

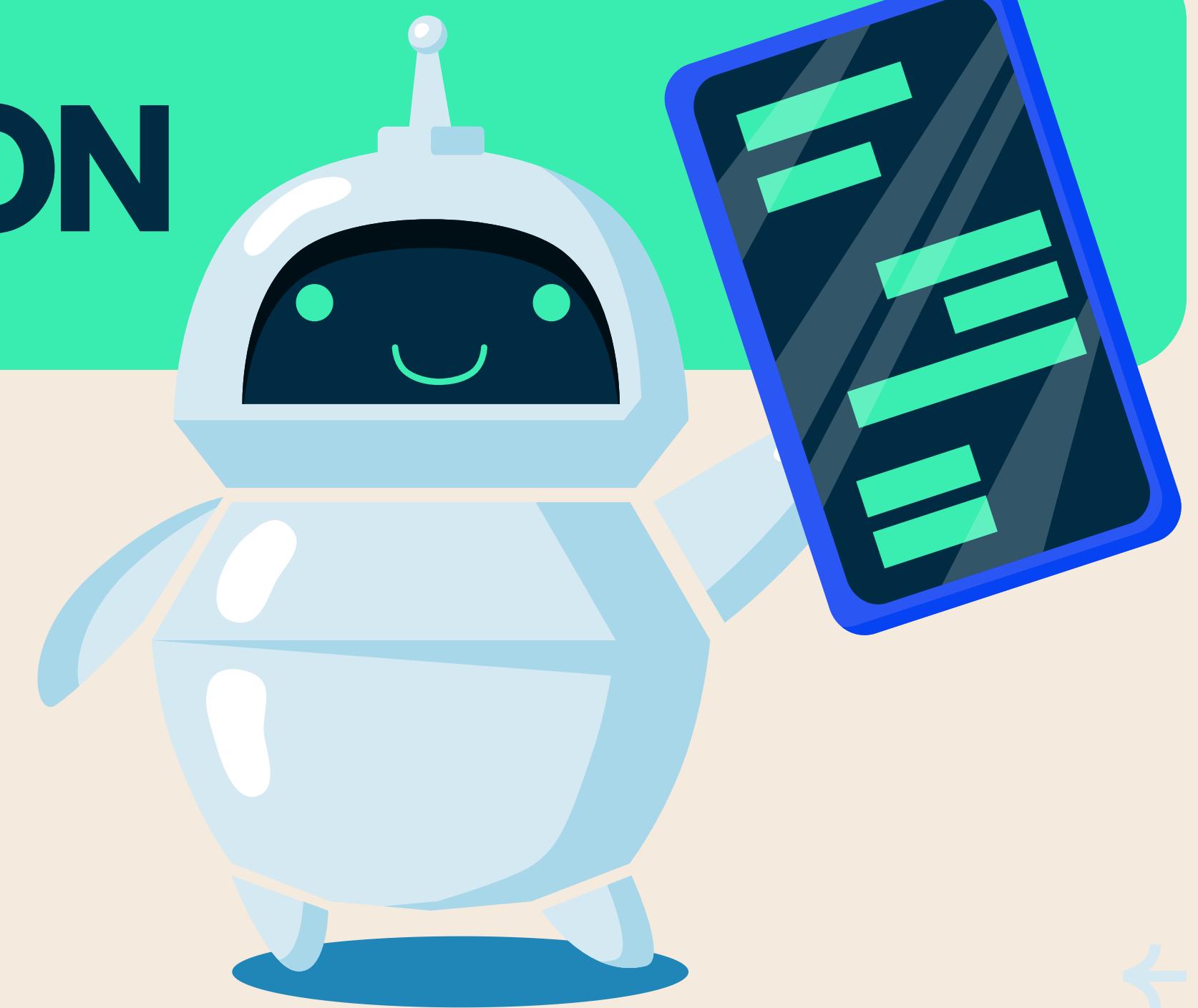
Eclipse



Github



SceneBuilder



RESULTS



- User home page
- Robot testing
- Session name edit
- Top3 ranked button action

User home page

Logout

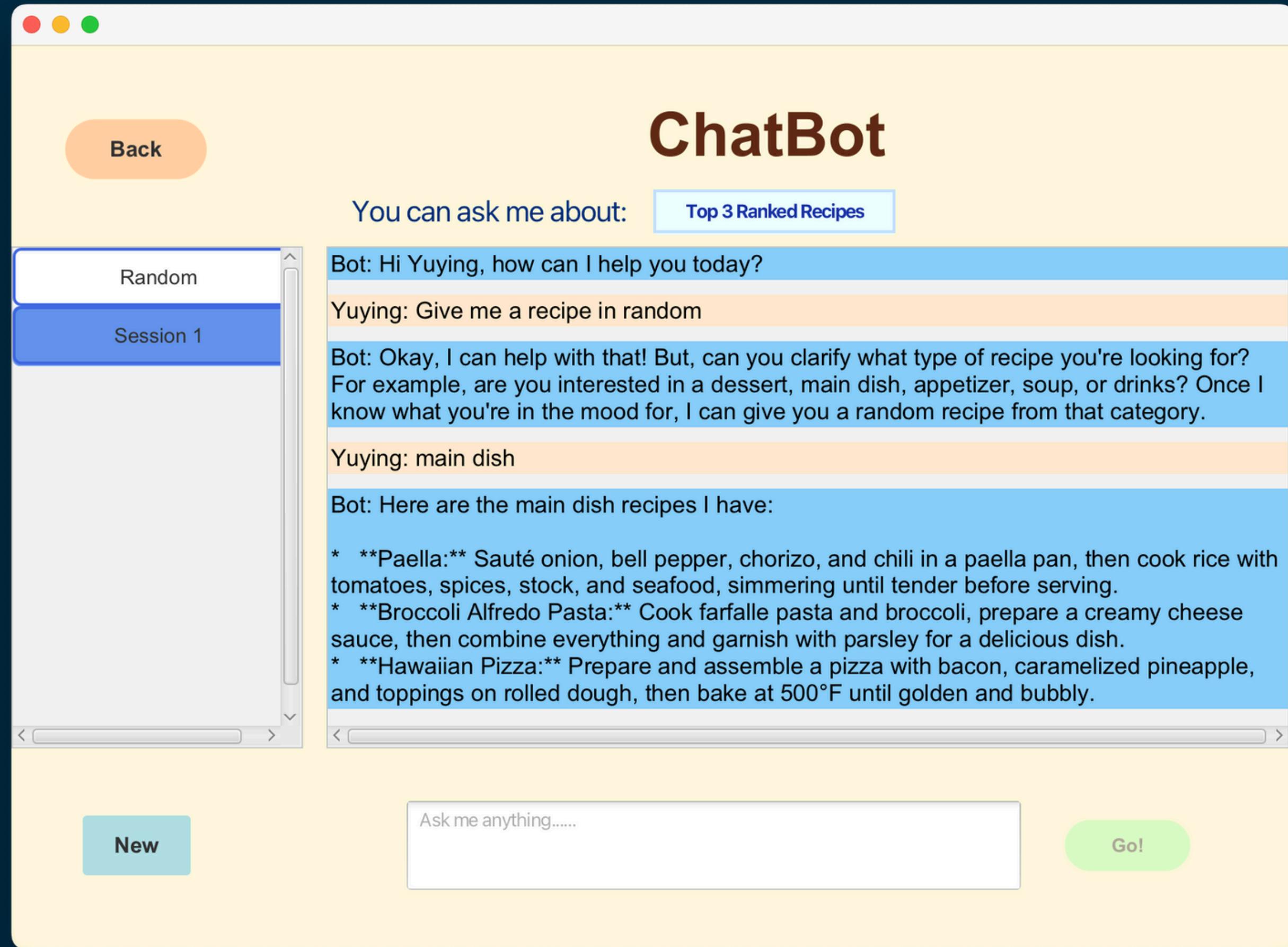
Welcome User

click

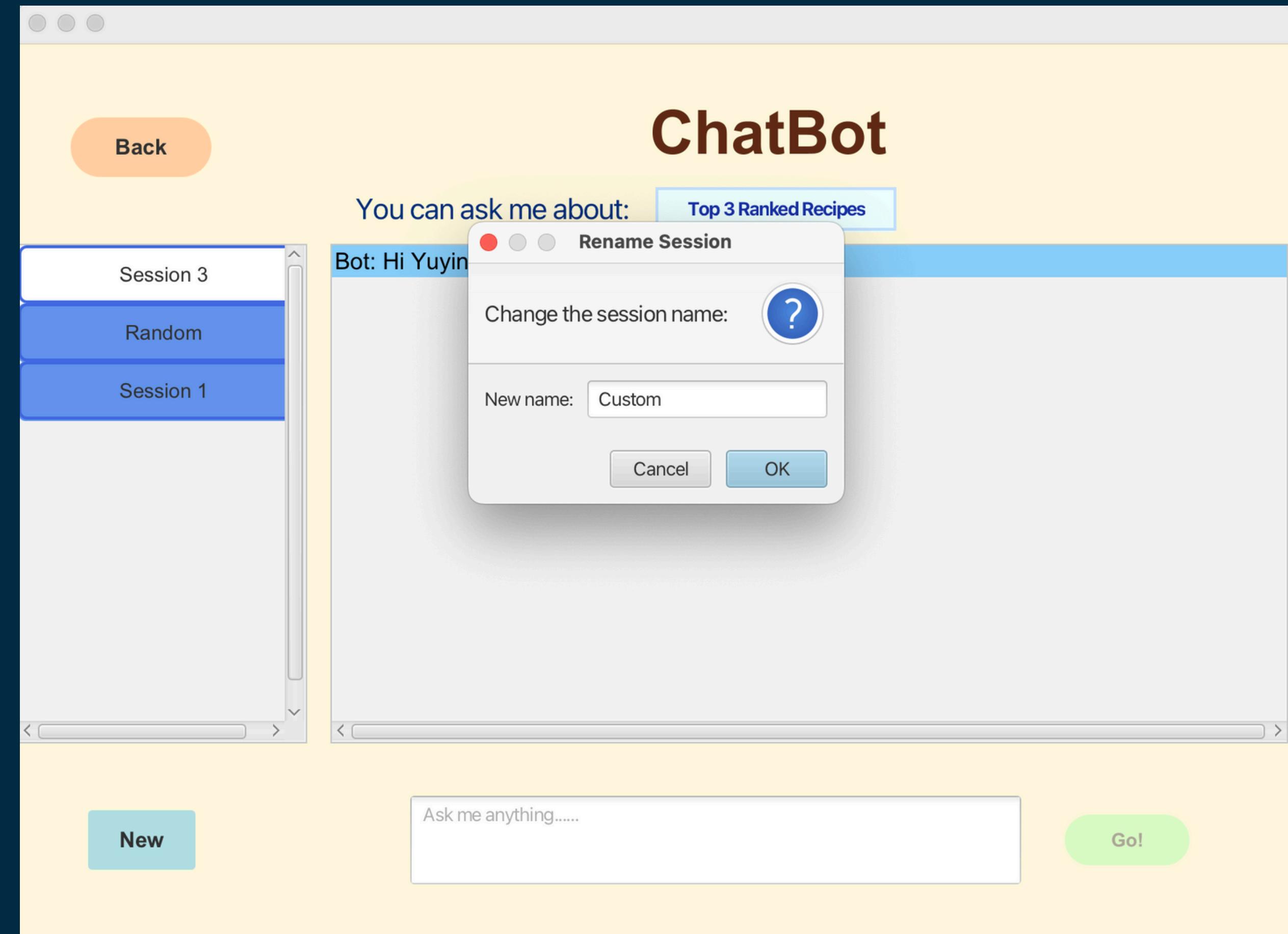
Create Recipe ChatBot Query Enter Ingredient Search

Baklava	Broccoli Alfredo ...	Hawaiian Pizza	Holiday Potatoes	Polish Potato So...
Sweet Lime Iced...	South Carolina S...	Eggplant Appeti...	Paella	Chocolate Lava ...
Tiramisu	Lemon Meringu...	Cheesecake	Apple Crumble	Panna Cotta
Strawberry Shor...	Creme Brûlée	Churros		

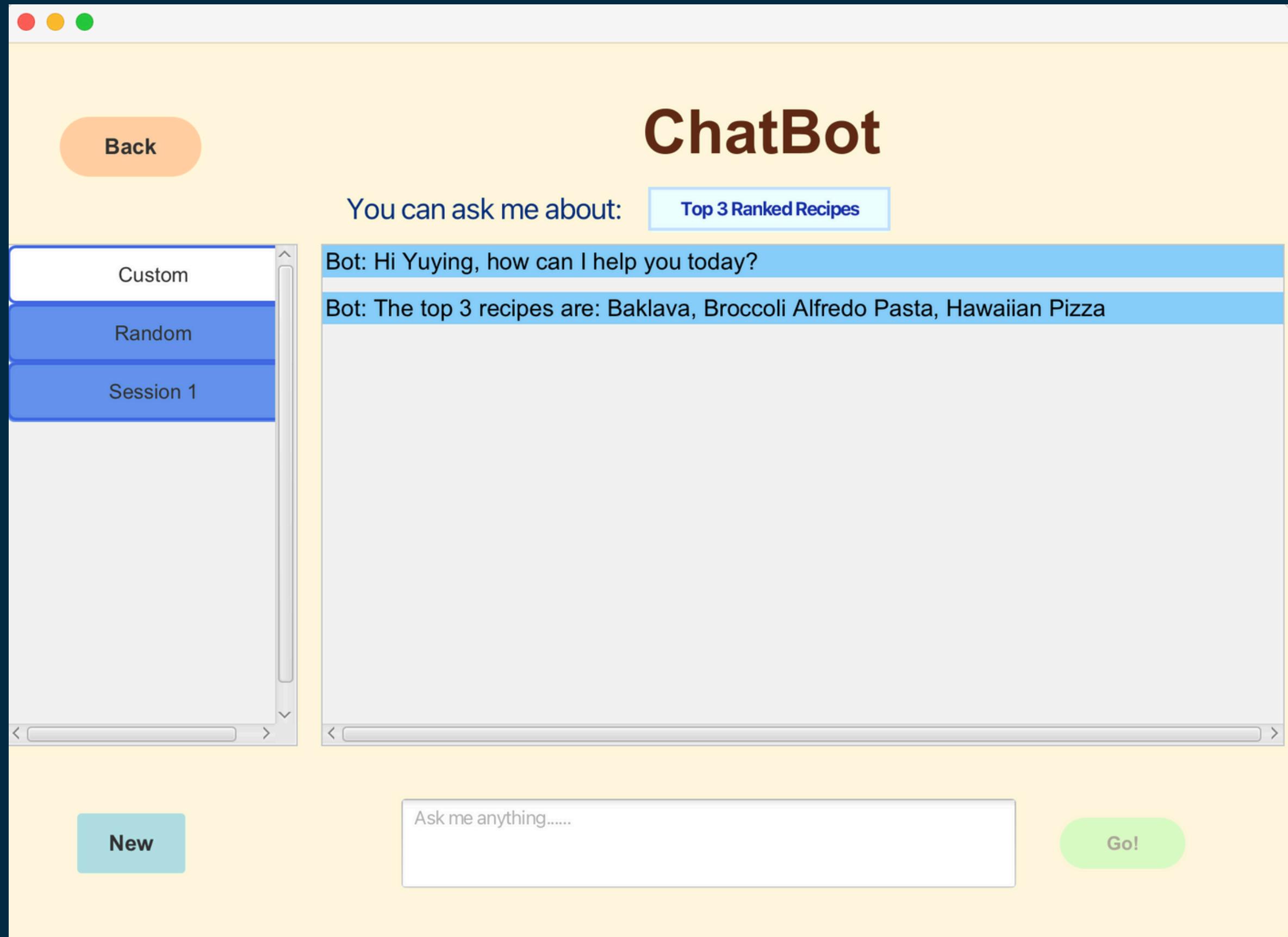
Robot testing



Edit session name



Top3 Ranked Recipes button

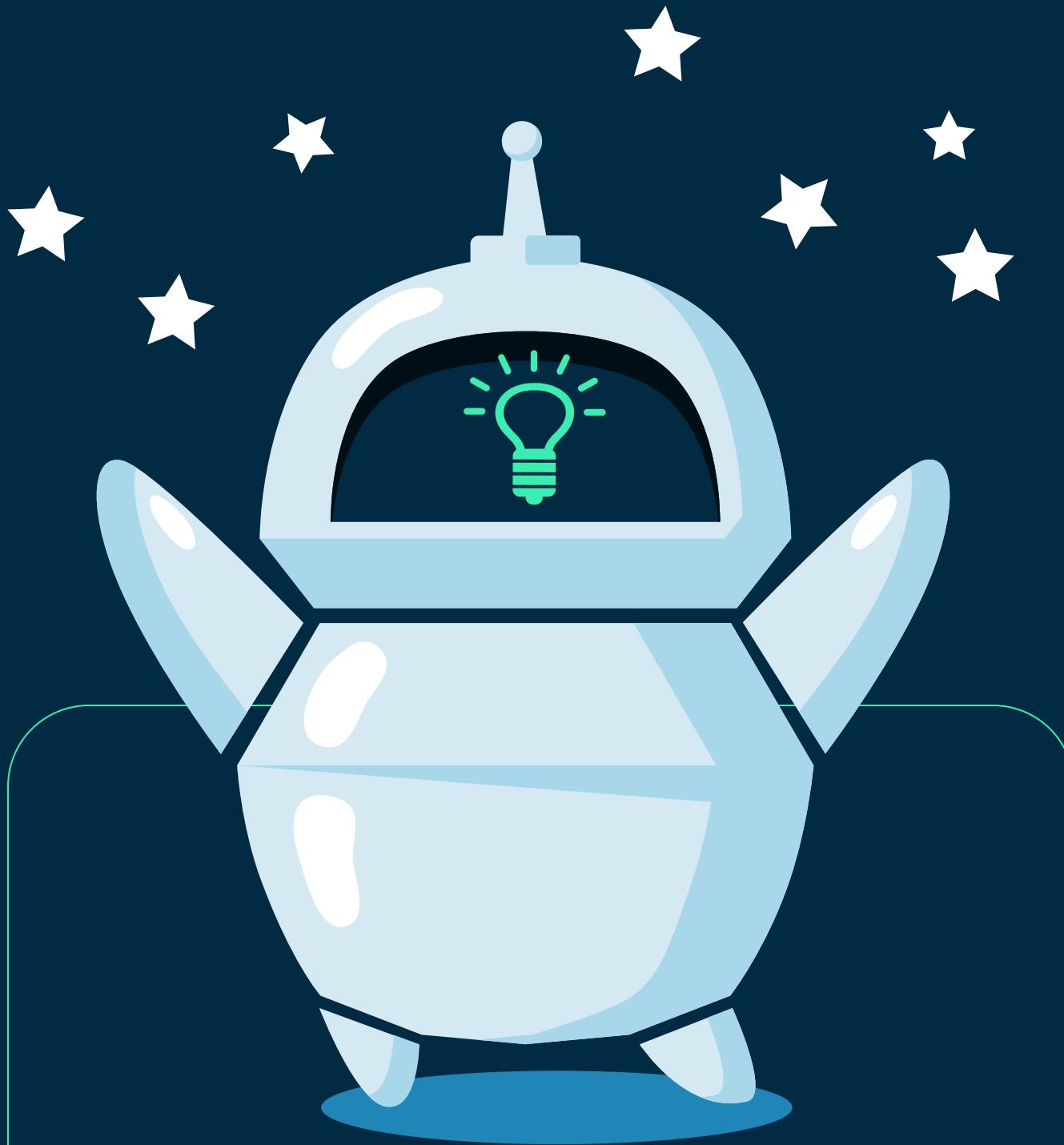


REFLECTION



- Priority Queue → Heap
- Database Deployment
- NLP techniques

CONCLUSIONS



Intent Detection with NLP

- Utilize advanced NLP techniques to analyze user needs, including parameters like recipe type, flavor, and ingredients.

Optimized Knowledge Base Retrieval

- Accurately detecting user intent allows us to retrieve the most relevant knowledge, ensuring efficient and focused responses while avoiding irrelevant information.



- Efficiency improvement
- Cost reduction

JOB ASSIGNMENT

Jiaye - UI design& controller development/ merge sort/ recursion/ queue
Qiuyue - List& ArrayList Design/ chat session & history management
Yuying - architecture design/ hashing/ knowledge base/ LLM API



THANK
YOU

→ DEMO