MealMate: From Food Cravings to Shopping Carts

CS194/294-196 LLM Agents

Jonas Li, Jasper Liu, Mats Wiig Martinussen, Emil Klovning, Jason Ji, Qinghe Zheng

SID: 3040802664, 3038111637, 3040788234, 3040759348, 3035562490, 3040860189

*Application track with 4 units of all members

Abstract

We introduce **MealMate**, a personalized food shopping assistant that seamlessly connects recipe generation with real-time ingredient availability. Using a multi-agent pipeline powered by large language models (LLMs), our end-to-end system creates recipes tailored to both user preferences and the inventory of nearby stores. MealMate comprises two key agents: the **Chef De Cuisine Agent**, which interprets user requests to develop customized recipes, and the **Line Cook Agent**, which verifies ingredient availability at local retailers.

Our solution is delivered through an intuitive web application, providing users with actionable shopping lists that align with their preferences and current store inventories. Initial tests show that this approach effectively handles dynamic grocery conditions, offering a practical tool for meal planning.

To validate its performance, we conducted both end-to-end and targeted tests. Two challenging test sets evaluated the system’s ability to accurately match recipe ingredients with store products—even when product names were not straightforward—and its skill in handling missing ingredients by either removing them or discarding the dish. After refining the agent setup, embedding models, similarity searches, and prompting strategies, MealMate achieved a score of 16/17 on the matching test and 12/12 on the ingredient handling test.

This project represents a novel fusion of personalized recipe recommendations and live store inventory management. By ensuring that generated recipes are always supported by available ingredients, we offer a more efficient and user-centric meal planning experience than traditional recipe generators by utilizing LLM agents.

1 Introduction

Recent advances in AI, particularly with models like GPT[1] and LLaMA[2], have revolutionized a range of applications, from natural language processing to vision-language tasks. These models, trained on vast amounts of data, have demonstrated impressive capabilities in various downstream tasks. However, despite their power, large language models (LLMs) like GPT often struggle to handle complex, real-world applications that require specialized knowledge and integration with dynamic, context-specific data, such as recipe generation based on ingredient availability in nearby stores.

A promising solution to this challenge is the use of agent-based architectures. This project introduces an agent-based approach to overcome these limitations, using specialized agents to handle specific components of the task—such as recipe generation and ingredient availability—allowing for more targeted and efficient responses. By combining the strengths of LLMs with the versatility of agent-based systems, this approach aims to create a more effective solution for personalized meal planning.

The goal of this project is to develop an end-to-end intelligent meal planning assistant capable of generating personalized recipes based on user prompts and user data, checking ingredient availability in nearby stores. This approach is driven by two key agents:

- Chef de Cuisine Agent: Translate user queries into recipe suggestions.
- Line Cook Agent: Look for local stores with the necessary ingredients and decide action when one or more ingredients are unavailable.

The system integrates real-time data on ingredient availability with recipe generation, offering users a seamless experience while minimizing the number of questions asked.

Contributions: the main contribution of our work is summarized as follows:

- Propose **MealMate**, a scalable agent-based system for recipe generation that leverages large language models, requiring no additional training. The system can recommend the best recipe based on user preferences.
- Created a user-friendly frontend website, designed to provide an intuitive interface for users, making meal planning and recipe generation practical and accessible in everyday life.

2 Related Work

2.1 Multi-Agent Collaborative Framework

Recent advancements in multi-agent frameworks like AutoGen[3] and LangChain[4] have highlighted the potential of leveraging large language models (LLMs) for complex task-solving through modular inter-agent collaboration. AutoGen enables customizable, conversable agents with flexible conversation patterns defined in natural language or code, while LangChain integrates LLMs with tools and memory for dynamic orchestration. Inspired by these open-source frameworks, MealMate adopts a multi-agent paradigm tailored for recipe generation, integrating LLMs with domain-specific functionalities to provide scalable and user-oriented solutions.

2.2 AI in Personalized recipe generation

Several studies have explored the intersection of AI and personalized recipe generation. Amiri et al. [5] developed a flexible meal-planning system tailored to individuals with diet-related health concerns, while platforms like Mealime [6] and Instacart [7] offer personalized meal plans that still require human oversight. More recently, FoodLMM [8], a multi-modal LLM food assistant, has demonstrated strong capabilities in ingredient recognition and recipe generation, laying the groundwork for more advanced systems. Additionally, emerging techniques such as Reflexion [9], Rah! [10], and StateFlow [11] will be leveraged to orchestrate the different stages of meal planning. Building on this, we propose an LLM agent that enhances recipe generation by integrating ingredient availability data from local grocery stores. This will enable more practical, real-time meal planning assistance.

While LLM-based recipe generators exist (e.g., FoodLLM) and many stores list ingredient availability online, no known solutions generate recipes directly from a store' s current inventory. This approach presents several challenges. A typical store inventory includes 30,000 to 60,000 items, making it impossible to include all product data in a single prompt. Therefore, instead of first providing context to an LLM about ingredient availability and then generating a recipe, we first generate a recipe, then look at availability based on matching ingredients with store inventory. Matching generic recipe ingredients with specific products is complex, as product names often contain extraneous details, such as brand names. Another challenge is how we can integrate user preferences into the recipe generation. Addressing these obstacles requires multiple specialized LLMs capable of abstracting and flexibly matching ingredients to products. In the following sections, we present our proposed solution, building upon the research and work discussed above.

3 Our Method

3.1 Overview

When a user requests a recipe, the query is first processed by the Chef de Cuisine Agent. The Chef de Cuisine agent then generates 3 recipes, taking into account user preferences, which in this part involves allergies. These 3 recipes are then presented to the customer. The customer chooses which of the three recipes they want to move forward with. The chosen recipe is then sent to the Line Cook Agent, which utilize several LLM agents to figure out the closest store that has all the ingredients available. Finally, the recipe, instructions and the closest store that have all the available ingredients are displayed to the user. The user can then select which of the ingredients they want to buy, and then proceed to checkout. The LLM agent set up in the Line Cook Agent is further elaborated below.

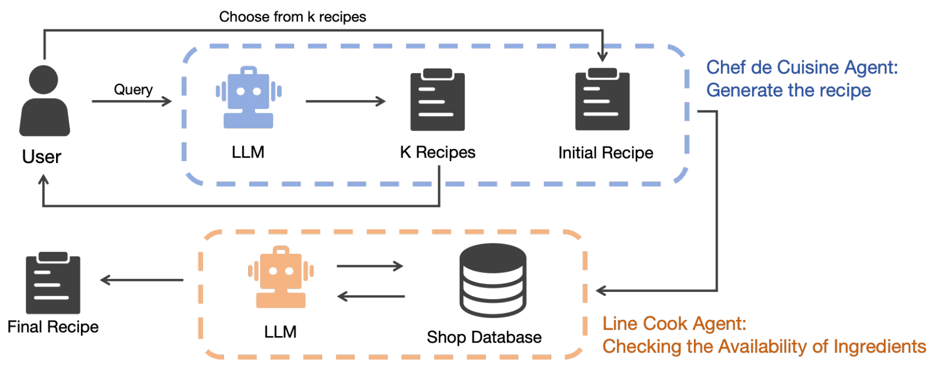


Figure 1: The overall workflow of MealMate

3.2 Line Cook Agent

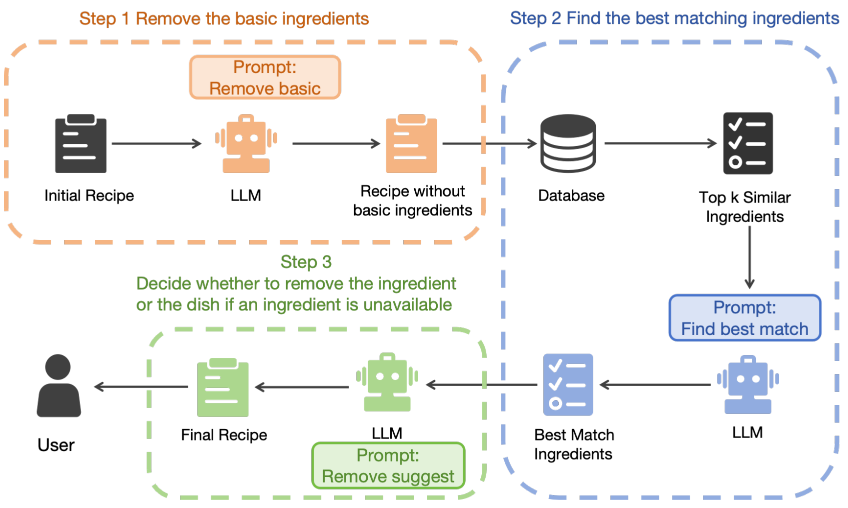

Step 1 Remove the basic ingredients
Prompt: Remove basic
Initial Recipe
LLM
Recipe without basic ingredients
Step 2 Find the best matching ingredients
Database
Top k Similar Ingredients
Prompt: Find best match
LLM
Best Match Ingredients
Step 3 Decide whether to remove the ingredient or the dish if an ingredient is unavailable
LLM
Prompt: Remove suggest
Final Recipe
User

Figure 2: Line Cook agent

The Line Cook Agent (see Figure 2) is designed to verify the availability of ingredients for a given recipe using a store’ s database. It begins by checking the store closest to the customer. If that store does not have all the essential ingredients, it moves on to the second closest store, and so forth. If no stores carry all the necessary ingredients, the agent informs the customer which items are unavailable, allowing the customer to consider alternative options. All large language models (LLMs) used in the Line Cook Agent are instances of gpt-4o-mini.

To determine ingredient availability, the process works as follows:

- **Identify Basic Ingredients:** First, an LLM is used to identify “basic” ingredients—items like salt, pepper, and oil. These basics are removed from the recipe under the assumption that most households already have them.
- **Match Ingredients to Store Products:** For each remaining ingredient, an LLM checks if the store carries products that match it. If multiple products are found, a formula that accounts for the user’ s preferences selects the best match.

Since the store’ s inventory database is too large for a single prompt, the system first narrows down candidates using similarity-based filtering. By computing cosine similarity scores between the ingredient’ s embedding and the database entries, the system identifies the top product matches before presenting them to the LLM for final selection. The similarity is calculated using the following formula:

$$\text{Cosine Similarity}(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}$$

Where $\mathbf{A} \cdot \mathbf{B}$ is the dot product of the vectors $\mathbf{A}$ and $\mathbf{B}$, and $\|\mathbf{A}\|$ and $\|\mathbf{B}\|$ are the magnitudes of the vectors.

We compute embeddings for the store inventory and the recipe ingredients using the pre-trained Sentence-BERT transformer model paraphrase-MiniLM-L6-v2. This model’ s small size and efficiency make it suitable for real-time applications. To reduce runtime, we pre-compute embeddings for all products in the stores’ inventories, so that only the recipe ingredients need to be embedded at runtime.

Next, we identify the top 10 closest product embeddings to each ingredient and provide these candidates to another LLM. The LLM then determines whether one or more of these candidates are valid matches. If no matches are found, the LLM outputs “None.”

If no suitable product can be matched for a given ingredient, a third LLM decides whether to remove that ingredient or discard the entire dish. This decision depends on whether the ingredient is essential. If the dish must be discarded, the system proceeds to check the next store. If none of the stores carry all essential ingredients, we inform the customer which ingredients are unavailable and prompt them to choose another dish.

When a store has all the necessary ingredients but offers multiple valid matches for one or more of them, we refine the selection by considering both semantic similarity and user-defined price preferences. After filtering for acceptable ingredients, we standardize prices (e.g., per pound) and normalize them, enabling a weighting scheme that balances similarity and cost. The user specifies the importance of price on a scale from 1 to 5: a 1 favors higher-priced options under the assumption of better quality, while a 5 emphasizes affordability. We then compute a final score for each candidate by combining its normalized price and similarity scores according to the user’ s preference. The product with the highest final score is chosen as the best match, ensuring that it not only closely resembles the requested ingredient but also aligns with the user’ s sensitivity to price.

By following this systematic approach—checking availability, considering substitutions, and respecting overall recipe feasibility—the agent effectively simulates a practical shopping decision-making process. As a result, it provides more realistic and user-centric suggestions that enhance the overall culinary experience.

Model	**Find Best Match**	**Replacement Suggestion**	**Testing Time**
gpt-3.5-turbo	0.71	0.42	10.6s
gpt-4o	0.82	0.83	19.63s
gpt-4o-mini	**0.94**	**0.83**	**11.28s**

Table 1: Test Accuracy for Different Models. This is based on our targeted tests were done before prompt optimizing, where we managed to further increase the scores.

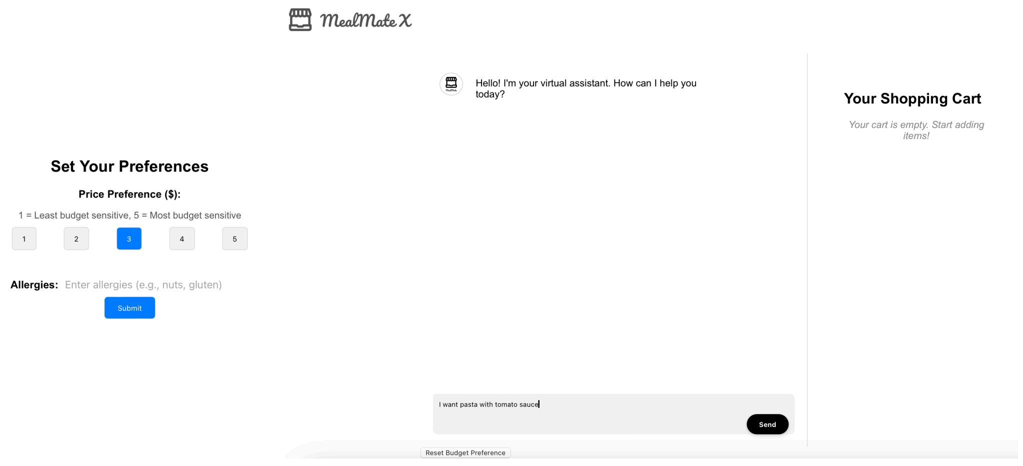
Based on the evaluation results, **GPT-4o-mini** is the optimal choice for our application. It achieves the highest accuracy in ”Find Best Match” (0.94) and matches GPT-4o in ”Replacement Suggestion” accuracy (0.83), significantly outperforming GPT-3.5-turbo in both tasks. Additionally, GPT-4o-mini demonstrates a faster testing time of 11.28 seconds compared to GPT-4o’s 19.63 seconds, making it more efficient without compromising performance. While GPT-3.5-turbo is the quickest at 10.6 seconds, its lower accuracy in both tasks (0.71 and 0.42) makes it unsuitable for our needs. GPT-4o-mini strikes the best balance between accuracy and speed, ensuring high-quality outputs while maintaining reasonable latency. These qualities make GPT-4o-mini an ideal model for real-time applications like MealMate, where both precision and efficiency are critical for providing a seamless user experience.

Algorithm 1 Line Cook Agent: Check Availability of Ingredients Across Multiple Stores

```
Require: Original Recipe $recipe$, Stores $\{S_1, S_2, \ldots, S_m\}$ sorted by proximity, Database entries
  per store, Number of Products $n\_products$, User Price Preference $p \in \{1, \ldots, 5\}$
  Load all store databases and extract product descriptions
  Load embeddings model and encode all product descriptions into embeddings
  Remove basic ingredients (e.g., salt, sugar, oil) from $recipe$ using LLM
  for each store $S_j$ in order of increasing distance do
    $recipe\_status \leftarrow$ ”All Ingredients Matched”
    for each ingredient $i$ in the modified $recipe$ do
      Calculate similarity scores between $i$ and the products in $S_j$'s database using cosine simi-
  larity
      Select the top $n\_products$ closest matches based on similarity scores
      Use LLM to verify which of these candidates are valid matches for $i$
      if no valid match is found then
        Use an LLM to decide whether to remove $i$ or discard the entire $recipe$
        if LLM suggests removing the dish then
          $recipe\_status \leftarrow$ ”Dish Removed”
          break (stop checking this store)
        else
          Mark $i$ as ”Ingredient Removed”
        end if
      else
        Identify if multiple valid matches exist for $i$
        if multiple matches then
          Compute a weighted score for each valid match that combines:
    - Cosine similarity
    - User price preference $p$ (use a price similarity formula)
          Choose the product with the highest weighted score as the best match
        end if
      end if
    end for
    if $recipe\_status$ == ”All Ingredients Matched” then
      return Selected products from $S_j$
    end if
  end for
  if $recipe\_status$ != ”All Ingredients Matched” then
    Inform user that none of the stores carry all essential ingredients
    Suggest the user consider another dish, listing the unavailable ingredients
  end if
```

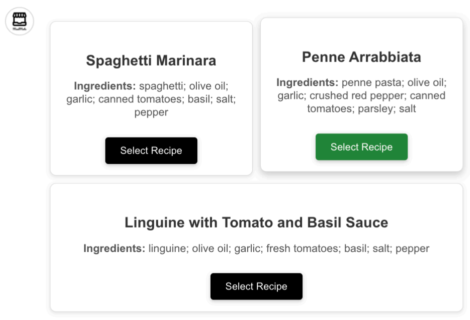
4 User Guide

When the user login on MealMate, they first need to select preferences, which currently includes price and allergies.



MealMate X
Set Your Preferences
Price Preference ($):
1 = Least budget sensitive, 5 = Most budget sensitive
1
2
3
4
5
Allergies: Enter allergies (e.g., nuts, gluten)
Submit
Hello! I'm your virtual assistant. How can I help you today?
I want pasta with tomato sauce
Send
Reset Budget Preference
Your Shopping Cart
Your cart is empty. Start adding items!

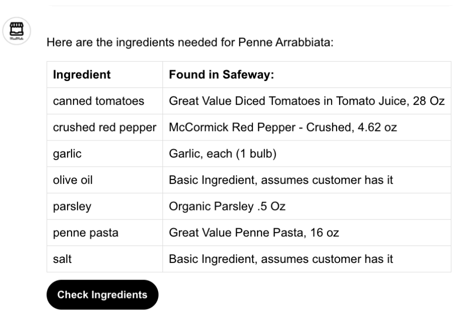
Then in the user interface, simply input the desired meal in the dialogue box and send it to the system. The prompt can be of any level of detail and specificity. Based on user preferences, the system will generate multiple recipe suggestions that correspond to the request. User can then select the recipe that best matches needs.



Spaghetti Marinara
Ingredients: spaghetti; olive oil; garlic; canned tomatoes; basil; salt; pepper

Penne Arrabbiata
Ingredients: penne pasta; olive oil; garlic; crushed red pepper; canned tomatoes; parsley; salt

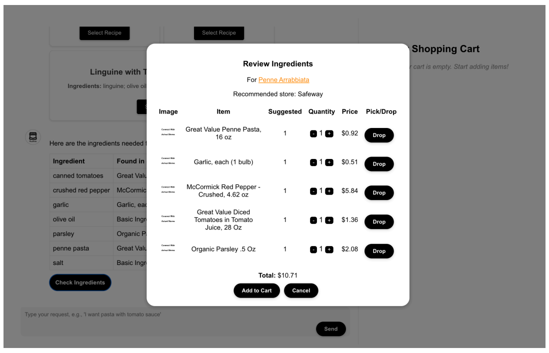
Linguine with Tomato and Basil Sauce
Ingredients: linguine; olive oil; garlic; fresh tomatoes; basil; salt; pepper



Here are the ingredients needed for Penne Arrabbiata:

Ingredient	Found in Safeway:
canned tomatoes	Great Value Diced Tomatoes in Tomato Juice, 28 Oz
crushed red pepper	McCormick Red Pepper - Crushed, 4.62 oz
garlic	Garlic, each (1 bulb)
olive oil	Basic Ingredient, assumes customer has it
parsley	Organic Parsley .5 Oz
penne pasta	Great Value Penne Pasta, 16 oz
salt	Basic Ingredient, assumes customer has it

After selecting the preferred recipe, the system will search the store's databases to find the closest store that has all the ingredients available. If no store has all the ingredients, the program asks the user to enter a new prompt as it was not possible to find ingredients for the current recipe. If one store has all the ingredients available, the user can click on the button "Check Ingredients". An interactive window where the user can decide the amount of each ingredient then pops up. Once the user clicks on "Add to Cart", a corresponding shopping list will be generated.


Select Recipe
Select Recipe
Shopping Cart
cart is empty. Start adding items!
Linguine with T
Ingredients: linguine; olive oil
Here are the ingredients needed f
Check Ingredients
Type your request, e.g., 'I want pasta with tomato sauce'
Send

Review Ingredients

For Penne Arrabbiata

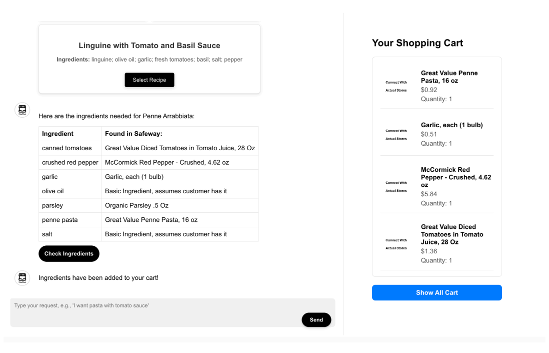
Recommended store: Safeway

Image	Item	Suggested	Quantity	Price	Pick/Drop
Connect With Actual Stores	Great Value Penne Pasta, 16 oz	1	- 1 +	$0.92	Drop
Connect With Actual Stores	Garlic, each (1 bulb)	1	- 1 +	$0.51	Drop
Connect With Actual Stores	McCormick Red Pepper - Crushed, 4.62 oz	1	- 1 +	$5.84	Drop
Connect With Actual Stores	Great Value Diced Tomatoes in Tomato Juice, 28 Oz	1	- 1 +	$1.36	Drop
Connect With Actual Stores	Organic Parsley .5 Oz	1	- 1 +	$2.08	Drop

Total: $10.71

Add to Cart Cancel

Ingredient	Found in
canned tomatoes	Great Valu
crushed red pepper	McCormic
garlic	Garlic, eac
olive oil	Basic Ingr
parsley	Organic P
penne pasta	Great Valu
salt	Basic Ingr



Linguine with Tomato and Basil Sauce

Ingredients: linguine; olive oil; garlic; fresh tomatoes; basil; salt; pepper

Here are the ingredients needed for Penne Arrabbiata:

Ingredient	Found in Safeway:
canned tomatoes	Great Value Diced Tomatoes in Tomato Juice, 28 Oz
crushed red pepper	McCormick Red Pepper - Crushed, 4.62 oz
garlic	Garlic, each (1 bulb)
olive oil	Basic Ingredient, assumes customer has it
parsley	Organic Parsley .5 Oz
penne pasta	Great Value Penne Pasta, 16 oz
salt	Basic Ingredient, assumes customer has it

Ingredients have been added to your cart!

Your Shopping Cart

Connect With Actual Stores
Great Value Penne Pasta, 16 oz
$0.92
Quantity: 1

Connect With Actual Stores
Garlic, each (1 bulb)
$0.51
Quantity: 1

Connect With Actual Stores
McCormick Red Pepper - Crushed, 4.62 oz
$5.84
Quantity: 1

Connect With Actual Stores
Great Value Diced Tomatoes in Tomato Juice, 28 Oz
$1.36
Quantity: 1

5 Conclusion

In this report, we introduced **MealMate**, an intelligent meal planning system designed to provide users with quick and efficient grocery recommendations, saving valuable time in their daily routines. By integrating user preferences into the recipe generation and ingredient selection processes, MealMate delivers more personalized and tailored results, enhancing the overall user experience.

The system leverages two specialized agents: the **Chef de Cuisine Agent**, which generates initial recipe suggestions from a database, and the **Line Cook Agent**, which verifies ingredient availability and suggests alternatives if needed. This streamlined workflow ensures that users receive actionable grocery lists that align with their needs and preferences.

Future developments will aim to refine ingredient substitution, expand preference-based recommendations, and integrate nutrition-focused features, further solidifying MealMate as a trusted companion for meal planning and grocery shopping.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[3] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. Autogen: Enabling next-gen llm applications via multi-agent conversation, 2023.

[4] Harrison Chase. Langchain, 2022. Accessed: 2024-12-09.

[5] Maryam Amiri, Juan Li, and Wordh Hasan. Personalized flexible meal planning for individuals with diet-related health concerns: System design and feasibility validation study. *JMIR Formative Research*, 7:e46434, 2023.

[6] Meal planning app for healthy eating - get it for free today! https://www.mealime.com/, n.d.

[7] Instacart. Instacart, n.d. Retrieved from https://www.instacart.com/.

[8] Yuehao Yin and et al. Foodlmm: A versatile food assistant using large multi-modal model. *arXiv preprint arXiv:2312.14991*, 2023.

[9] Noah Shinn and et al. Reflexion: Language agents with verbal reinforcement learning. *arXiv preprint cs.AI/2303.11366*, 2023.

[10] Yubo Shu and et al. Rah! recsys - assistant - human: A human-centered recommendation framework with llm agents. *IEEE Transactions on Computational Social Systems*, 2024.

[11] Yiran Wu and et al. Stateflow: Enhancing llm task-solving through state-driven workflows. *arXiv preprint arXiv:2403.11322*, 2024.