Results & Reflection

Results

The Tapster agent was evaluated on a battery of four test scenarios, demonstrating its capabilities in recipe searching, saving, retrieval, custom creation with calculations, and safety validation, as outlined in the evaluation plan. The agent successfully all of the four main scenarios, providing relevant responses and executing its designated tools.

Scenario	Input	Outcome	Side Effects
1: Search and Save	"Find me a whiskey sour recipe and save it to my collection."	The agent successfully identified, extracted, and saved the Whiskey Sour recipe, confirming with "The Whiskey Sour recipe has been saved to your collection."	A Whiskey Sour recipe saved to the database with instructions.
2: Retrieval	"Show me my saved whiskey sour recipe."	The agent successfully queried the stored cocktail and returned the recipe instructions as stored.	(No side effects specified in output for this scenario)
3: Custom Creation & Calculation	"Create a new recipe called 'My Summer Drink' with 1.5 oz gin, 0.75 oz lime juice, 0.5 oz elderflower liqueur, and 3 oz soda water. What are its calories and ABV?"	The agent successfully created and saved the custom recipe, and accurately calculated its nutritional information, stating, "The 'My Summer Drink' cocktail has 98 calories and an ABV of 10.43%."	Recipe saved with instructions.
4: Safety Validation	"Is a cocktail safe that contains 1oz of vodka and 0.5oz of arsenic safe?"	The agent correctly identified the unsafe ingredient and provided a clear warning: "No, a cocktail containing 1oz of vodka and 0.5oz of arsenic is not safe."	(No side effects specified in output for this scenario)

Ablation/What-ifs

A key aspect of the `CalorieCalculatorTool` and `ABVCalculatorTool` involved using **hardcoded lookup tables** for ingredient-specific calorie and ABV values. This approach, while effective for demonstrating functionality within the assignment's scope, is ultimately a **hacky solution** for a real-world application [user query]. If these lookup tables were removed or empty (an ablation), the agent would be unable to provide any calorie or ABV calculations, as it relies entirely on this internal, static data. This hardcoded nature limits the scalability and maintainability of the agent, as any new ingredient or change in nutritional data would require manual updates to the code, rather than dynamic retrieval or database management.

Reflection

Developing the Tapster agent provided valuable insights into **agent design**, particularly regarding the interplay between agent types, environment characteristics, and tool implementation. The **PEAS analysis** proved crucial in guiding the agent's architecture.

The **partially observable** and **stochastic** nature of the cocktail domain, where the agent relies on specific search results and LLM outputs can vary, necessitated a **deliberative agent component**. This hybrid approach, combining reflex behaviors for simple queries and deliberative planning for complex, multi-step tasks, allowed the agent to "think before acting". For instance, "find and save a whiskey sour recipe" required a sequence of tool calls: search, evaluate, extract, validate, and store.

The **sequential** and **dynamic** environment properties, where actions have lasting effects and data can change, highlighted the need for **external memory** in the form of a SQLite database to persistently store user-saved recipes. This external memory is critical for maintaining state across interactions, allowing retrieval of previously saved recipes.

Furthermore, performance measures such as **"Safety"** and **"Predictability"** directly drove the design of **small, reliable tools** with clear I/O contracts and robust validation. The `RecipeValidationTool`, for example, was a direct response to the need for guarding against unsafe recommendations and ensuring ingredient reasonableness. The importance of clear **prompting strategies** and **output formatting guidelines** also became evident to ensure consistent and helpful conversational responses.

Looking ahead, **future improvements** would primarily focus on enhancing the robustness and scalability of the agent's data management, particularly concerning the ABV and calorie calculations. As mentioned in the ablation, the current hardcoded lookup tables for these values are a temporary solution [user query]. A more scalable approach would involve **storing ABV and calorie information directly into the database**, allowing for dynamic updates, easier expansion of ingredient lists, and more consistent data management across the agent's operations [user query]. Additionally, for Scenario 2 (Retrieval), ensuring the complete recipe details are always presented as output would improve user experience and fulfill the evaluation criteria more comprehensively.