Just eat takeaway

Research Document

Project Team

Jorzinho Martina

Baris Buba

Alaa Bolbol

Tamas Wijk

Laurens Ton Janssen

Research Team:
Jorzinho Martina
Baris Buba
Laurens Janssen

Research Team:
Alaa Bolbol
Tamas Wijk

Contents

1. Introduction

1.1 Context

Takeaway is a platform where restaurants and customers can easily interact with each other. Customers order food, restaurants prepare the meals, and delivery drivers bring them to their destination. The process is simple and convenient, making it easy for people to get meals without cooking. However, making a healthy choice isn't always as easy.

When opening the app or website, customers are immediately shown popular options, which often include fast food like burgers, pizza, and fried chicken. These meals are tasty and familiar, but they aren't always the healthiest choice. Without actively searching, healthier alternatives are not always visible and many users don't take the time to look for them. As a result, they might develop habits of ordering less nutritious meals without even realizing it.

This project focuses on addressing that challenge by introducing an AI-powered assistant that helps users make better food choices without taking away their freedom to choose what they like. The AI will offer subtle suggestions based on user preferences, encouraging healthier eating habits without making users feel restricted.

1.2 Problem definition

The Takeaway app makes ordering food super easy, but the meals that appear first are mostly pizza, burgers and other fast food. Because these dishes are right at the top, many people never scroll down far enough to see healthier options like salads, grain bowls or grilled dishes. This is a form of *visibility bias*: what is easiest to see feels like the easiest choice.

When someone is hungry or in a hurry, that bias becomes even stronger. Most users simply tap the familiar, tasty item and move on. Repeating this pattern turns it into a habit of choosing meals that are high in calories, salt and sugar. Over time, that habit can push out vegetables, whole grains and other nutrient-rich foods, raising the risk of weight gain and diet-related health issues.

1.3 Research Goal

Develop and evaluate an AI-powered conversational assistant that unobtrusively promotes healthier meal choices within the Takeaway ordering flow. The assistant should:

- Surface healthier menu items that match each user’s tastes and dietary restrictions.
- Provide concise, evidence-based nutrition reasoning for its suggestions.
- Respect user autonomy by framing suggestions as optional nudges rather than prescriptions.

Success will be measured through improvements in the nutritional quality of orders during user testing, positive user feedback, and strict adherence to dietary constraints. So, the most results will come from testing, we need to give the user the “app” and see if our AI-assistent will guide them slowly towards a healthier option.

1.4 Research Questions

Main-Questions

How can we create a food chatbot that will make proper, well informed food recommendations to its users?

Sub-Questions

1. Which technologies will be used to build this chatbot?
2. Which nutrition framework should be adopted to ensure healthier lifestyles?
3. How can we ensure that the model avoids recommending food that goes against a user's preferences? (Pork to halal users, meat to plant-based users)
4. How can we mitigate model hallucination?
5. How will we measure the model's performance?

1.5 Research Method

Question	Method	Strategy	Answered By
Which technologies will be used to build this chatbot?	Library	Available product analysis Community research	Tamas
Which nutrition framework should be adopted to ensure the chatbot promotes healthier lifestyles?	Library	Literature study	Laurens
How can we ensure that the model avoids recommending food that goes against a user's preferences? (Pork to halal users, meat to plant-based users)	Lab Field	Model Validation Focus group? Interview Task analysis	Alaa
How can we mitigate model hallucination?	Lab Library	Usability Testing Literature study	Baris
How will we measure the model's performance?	Lab	Model Evaluation	Zinho

2. Research

2.1 Which technologies will be used to build this chatbot?

In choosing the tools for our standalone chatbot we explored several alternatives. We compared front-end frameworks such as Angular, Vue and React, and found that React, paired with the fast build tool Vite, offers the quickest development cycle while remaining familiar to most web developers at Just Eat Takeaway. React's component model lets us compose chat bubbles, input fields and other UI elements in a clean, reusable way; Vite gives near-instant hot-reloads and optimised production bundles.

For the back-end and conversational intelligence we evaluated Node.js and Python ecosystems. We selected Python together with the LangChain framework. Python's mature AI libraries (like PyTorch and Hugging Face) make model integration straightforward, and LangChain supplies ready-made building blocks for prompt handling, retrieval-augmented generation, memory and tool usage. This lets us focus on food-specific business logic instead of low-level orchestration.

All code lives in GitHub

This stack is lightweight enough for rapid prototyping but scales smoothly when we plug the chatbot into Just Eat Takeaway's production infrastructure later.

2.1.1 Conclusion

In conclusion, the chatbot will be built using a modern and efficient tech stack: React with Vite for the front-end, and Python with the LangChain framework for the back-end and conversational intelligence. This combination was chosen for its speed, scalability, and alignment with existing developer expertise. React and Vite provide a fast, modular, and developer-friendly UI foundation, while Python and LangChain offer powerful tools for integrating AI models and managing conversational flows. This setup allows for rapid prototyping today and seamless integration into Just Eat Takeaway's production systems in the future.

2.2 Which nutrition framework should be adopted to ensure the chatbot promotes healthier lifestyles?

In our project, we initially explored different nutrition frameworks to define what we mean by "healthier." After comparing several options including the NHS traffic light system, WHO guidelines, and Nutri-Score. We decided to base our definition on the Dutch Voedingscentrum (Nutrition Centre) and its Schijf van Vijf model.

We chose this framework because it offers a clear and objective set of factors: meals should contain enough vegetables, whole grains, limited saturated fat, and not too much salt or sugar. This gives us a concrete foundation for defining and promoting healthier choices.

Other frameworks had downsides that made them less suitable for our purpose. The NHS traffic light system is designed mainly for packaged products, the WHO guidelines are too general, and Nutri-Score lacks transparency in how the final score is calculated. For our nudging strategy within the Thuisbezorgd environment, the Voedingscentrum approach is the most practical, recognizable, and evidence-based.

The information given to the chatbot will be looking like this: Healthy meal classification based on Schijf van Vijf (Dutch Nutrition Guidelines), adapted for fast food context:

- **Preferred ingredients:** vegetables (e.g. lettuce, tomato, onion, cucumber, spinach), whole grain products (e.g. whole wheat buns, whole grain wraps, brown rice), legumes (e.g. beans, lentils, chickpeas), unsalted nuts, fresh fruit, lean proteins (e.g. grilled chicken, fish, tofu).
- **Limit these:** cheese, low-fat dairy (e.g. milk or yogurt-based sauces), white bread or refined carbs (allowed, but not preferred), plant-based meat alternatives (only if not highly processed).
- **Minimize or avoid:** red or processed meat (e.g. beef patties, bacon, sausage), fried foods (e.g. fries, fried chicken, onion rings), high-sugar items (e.g. desserts, soft drinks, milkshakes, sauces with sugar), salty, fatty toppings (e.g. mayo, cheese sauce, creamy dressings), alcohol, energy drinks.
- **Portion control:** total energy per meal should stay within healthy daily limits (approx. 600–800 kcal for main meal); limit salt (<2g), saturated fat (<10g), and added sugars (<25g) per meal.
- **Scoring suggestion:** a meal is considered healthier when it mostly consists of ingredients within the guidelines and contains little or nothing outside the Schijf van Vijf
-

2.2.1 Conclusion

After evaluating several nutrition frameworks, including the NHS traffic light system, WHO guidelines, and Nutri-Score, we chose to base our approach on the *Schijf van Vijf* developed by the Dutch Nutrition Centre. This decision is grounded in its scientific foundation, government endorsement, and practical clarity.

The *Schijf van Vijf* outlines what a healthy meal should include: plenty of vegetables, whole grains, plant-based proteins, healthy fats, and limited amounts of salt, sugar, and saturated fat. Other models were less suitable for our context. The NHS system focuses primarily on packaged foods, WHO guidelines are too broad for direct implementation, and Nutri-Score lacks transparency in how the final rating is calculated.

For our AI assistant, which operates within the fast food environment of Just Eats Takeaway, the *Schijf van Vijf* provides a clear and reliable structure to define and encourage healthier choices.

2.3 How can we ensure that the model avoids recommending food that goes against a user's preferences? (Pork to halal users, meat to plant-based users)

Collecting and Storing User Dietary Preferences:

The system should explicitly solicit and record each user's dietary restrictions (e.g. vegan, vegetarian, halal, kosher, allergies) in the user profile

Tagging and Structuring Menu Data:

All menu items must be annotated with rich metadata to enable precise filtering. This includes explicit dietary tags (e.g. Vegetarian, Vegan, Gluten-Free, Halal, etc.) and ingredient or allergen lists. For instance, many restaurant POS systems let operators tag dishes with abbreviations (e.g. "V" for vegetarian, "VG" for vegan, "GF" for gluten-free) and even allergen indicators.

In addition to manual tags, the menu database can store key ingredients (e.g. "contains pork", "contains dairy") or use automated analysis. For example, AI-powered menu digitization tools can identify ingredients from images or descriptions and flag dishes accordingly.

Pre-Filtering Candidate Items:

Before invoking an LLM for suggestions, the chatbot should filter the menu dataset based on the stored user profile. In practice this means querying only items that match the user's dietary tags and do not contain forbidden ingredients.

Prompt Engineering to Enforce Constraints:

Inside the language model, prompts must explicitly encode the dietary rules. The system prompt or the user's query should clearly instruct the LLM to include only compliant items. For example, one can prepend a directive like: "The user is [VEGETARIAN/VEGAN/HALAL/etc.]. Do not recommend any dishes that violate this restriction." Similarly, positive framing helps (e.g. "Recommend 5 vegan mains from the menu"). This technique mirrors "alignment specifications" used in AI safety; for instance, researchers suggest rules such as "Responses must respect vegetarian and vegan dietary choices" in the instruction set.

Post-Processing Safety Checks:

After the LLM generates its suggestions, a final validation step should catch any oversights. The system should automatically scan the output list for forbidden content: for example, checking if any recommended dish's name or description includes disallowed terms ("bacon", "pork", "gelatin", etc.) or if it appears in the menu database under a disallowed category. Any such item must be removed or replaced before showing the answer to the user.

In all cases, the chatbot should avoid recommending outright forbidden items. Clear communication and the option to adjust settings help maintain user trust when exact matches are not available.

2.3.1 Conclusion

In conclusion, ensuring the chatbot avoids recommending food that conflicts with a user's dietary preferences involves a multi-layered strategy. First, user-specific restrictions must be explicitly collected and stored in their profile. Next, all menu items need to be accurately tagged with dietary metadata and ingredient details, enabling precise filtering. Before querying the language model, the system should pre-filter the menu based on the user's restrictions. Within the model, well-crafted prompts reinforce these constraints by clearly stating the user's dietary needs. Finally, a post-processing validation step catches any remaining issues by scanning for non-compliant items. This end-to-end approach—spanning data collection, filtering, prompt design, and safety checks—ensures the chatbot consistently respects user preferences and maintains trust.

2.4 How can we mitigate model hallucination?

We looked into several options to combat model hallucinations. Currently we are going to make good use of the prompts to encourage the behavior of the model as we want it to. On top of that we will use deepeval to measure the model's performance to check if it responds correctly, uses the right language, hallucinates and so on. Based on that we can see how close the current model's behavior is to our expected behavior and then we can tweak it further if necessary. On top of that, we try to finetune the model on data that we worked on and verify it to be good for use.

Other methods have been looked into as well, like bespoke Mini checks where we check if the model response verifies the information coming from the document. If the claim matches, then the respond will be put out, otherwise it will not.

We also looked into multi-step verification with red teaming. Basically, a second LLM model (same or a different one) will act as a critical check that judges the answers given by the first model. So, by double checking the answer it will reduce hallucinations.

Due to the limited time, we can’t apply most ways to reduce hallucinations. However, it is clear that we thoroughly did our research to know what methods and tools are out there to get the most out of the model. If we do get the extra time and hands, we will be able to apply more options to the model to make sure that hallucinations occur less.

2.4.1 Conclusion

We’ve explored several ways to reduce hallucinations in our language model. Right now, we’re focusing on using carefully written prompts to guide and encourage the model's behavior in the desired direction. We're also using DeepEval to check if the model gives correct answers, uses the right language, and avoids making things up. This helps us see how close the model's behavior to what we are aiming for, so we can adjust it if needed.

Additionally, we're fine-tuning the model with data we've verified to be good. We've also looked into other methods, like checks to verify information and using a second model to double-check answers.

Although we can't implement all these methods right now due to time constraints, we've done thorough research to understand the available tools and techniques. If we get more time and resources, we'll apply more of these options to further reduce hallucinations.

2.5 How will we measure the model's performance?

We will measure the performance of our AI chatbot using the DeepEval framework, which allows for structured and concise evaluation of language models, particularly in Retrieval-Augmented Generation (RAG) settings which our chatbot is mainly built on. DeepEval enables both automated and “human in the loop” assessments, offering fine tuned control over custom metrics and prompt-based evaluation. We are using ollama as our LLM and since this is a widely used LLM by many we trust that it has been thoroughly tested, therefore we will mainly be focused on testing *RAG-Specific Metrics.*

We will evaluate our chatbot using the following metrics:

RAG-Specific Metrics

- **Answer Relevancy**: Measures how well the generated response answers the user's question. This helps us assess if the chatbot stays aligned with the user's intent.
- **Faithfulness**: Evaluates whether the chatbot's response is grounded in the retrieved context and avoids hallucination or fabrication.
- **Contextual Precision**: Determines the proportion of the retrieved context that is actually relevant to the user's question.
- **Contextual Recall**: Measures how much of the available relevant context was successfully retrieved and used.
- **Contextual Relevancy**: Assesses the overall utility and alignment between the retrieved context and both the question and response.

Why DeepEval?

We chose **DeepEval** over other evaluation frameworks (such as LangChain Evals or TruLens) for several reasons:

- **Built-in support for RAG evaluation**: DeepEval includes out of the box evaluators specifically tailored for faithfulness, context precision/recall, and hallucination which align directly with our use case.

- **Customization & Prompt-Based Flexibility**: It allows custom evaluation prompts and scoring logic, which is essential when dealing with nuanced behaviors like dietary bias or soft nudges in decision-making.
- **Human + Automated Evaluation**: DeepEval combines automated scoring with human overrides, giving us both speed and accuracy when judgment is required.
- **Extensibility**: We can easily log and score outputs across all test cases and integrate results into the testing pipeline for continuous evaluation.

By leveraging DeepEval, we aim to conduct a thorough evaluation of our chatbot, ensuring it meets our stakeholders' standards for accuracy and relevance. (To view the actual test cases, we ran on our chatbot please refer to the ***"JET - Test Report"***)

2.5.1 Conclusion

In conclusion, our approach to evaluating the chatbot's performance is designed to be both rigorous and reflective of its real-world purpose: helping users make more conscious food decisions. By using the DeepEval framework, we'll be able to test the chatbot across a diverse range of metrics from RAG-specific metrics like answer relevancy and faithfulness to broader concerns like bias, hallucination, and summarization quality.
DeepEval provides the structure, customization, and balance between automation and human judgment necessary to evaluate unwanted behaviors, especially in a domain where health, culture, and personal choice are important. Its dedicated support for Retrieval-Augmented Generation and flexible integration capabilities makes it particularly well-suited for our use cases.

This evaluation strategy ensures that our chatbot is not only technically sound but also responsible, inclusive, and aligned with our goal of promoting healthier, user-respecting food choices.

3. Conclusion

How can we create a food chatbot that will make proper, well-informed food recommendations to its users?

To build a food chatbot capable of delivering proper, well-informed recommendations, we need a thoughtful integration of the right technologies, frameworks, and safeguards. Our chosen tech stack React with Vite on the front-end and Python with LangChain on the back-end provides a fast, scalable foundation that supports both development efficiency and future integration with Just Eat Takeaway's systems. For the chatbot's intelligence layer, LangChain enables robust prompt management, retrieval-augmented generation, and memory, all essential for contextual, user-aware conversations.

Health-conscious recommendations are guided by the Dutch *Schijf van Vijf* nutrition framework, which offers a scientifically grounded, practical approach for identifying healthier food options within the fast food context. By embedding these dietary standards into the recommendation logic, the chatbot can nudge users toward better choices without being prescriptive.

To ensure alignment with individual dietary needs and restrictions (e.g., vegan, halal, gluten-free), we apply a multi-layered filtering approach. This includes explicit preference collection, structured menu tagging, LLM prompt constraints, and post-processing safety checks. This combination helps the chatbot consistently avoid inappropriate recommendations and maintain user trust.

Hallucination is addressed through carefully engineered prompts, model fine-tuning, and evaluation using *DeepEval*, which allows us to measure relevance, faithfulness, and contextual accuracy. Though time constraints limit the application of more advanced mitigation techniques like multi-step validation or red teaming, the foundation is in place for ongoing improvements.

Lastly, chatbot performance is continuously evaluated with DeepEval's RAG-specific metrics, combining automation with human oversight to ensure reliability, accuracy, and cultural sensitivity in food recommendations.

In summary, a successful food recommendation chatbot requires more than just AI it demands a reliable tech stack, a credible nutrition model, robust preference handling, hallucination safeguards, and a rigorous evaluation process. By combining all these elements, we can deliver a chatbot that is not only technically sound but also user-centric, trustworthy, and aligned with Just Eat Takeaway's mission to support healthier and more personalized food choices.

4. References

Confident-Ai. (n.d.). *GitHub - confident-ai/deepeval: The LLM Evaluation Framework*. GitHub. https://github.com/confident-ai/deepeval

GeeksforGeeks. (2024, September 13). *Evaluation metrics for RetrievalAugmented Generation (RAG) Systems*. GeeksforGeeks. https://www.geeksforgeeks.org/evaluation-metrics-for-retrieval-augmented-generation-rag-systems/

Gonzalez, L. (2025, March 5). *LLM Evaluation Frameworks: Head-to-Head Comparison*. Comet. https://www.comet.com/site/blog/llm-evaluation-frameworks/

Huang, J. (2025, March 18). Evaluating Large Language Model (LLM) systems: Metrics, challenges, and best practices. *Medium*. https://medium.com/data-science-at-microsoft/evaluating-llm-systems-metrics-challenges-and-best-practices-664ac25be7e5

AlfaPeople. (2024, December 13). *The importance of prompt engineering in preventing AI hallucinations*. AlfaPeople Global. https://alfapeople.com/importance-of-prompt-engineering-preventing-ai-hallucinations/?utm_source=chatgpt.com

Goortani, F. (2025, January 7). Strategies, patterns, and methods to avoid hallucination in large language model responses. *Medium*. https://medium.com/%40FrankGoortani/strategies-patterns-and-methods-to-avoid-hallucination-in-large-language-model-responses-81a871987d96

Dutta, N. (2025, January 24). *A Guide on Effective LLM Assessment with DeepEval*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2025/01/llm-assessment-with-deepeval/?utm_source=chatgpt.com

FONTYS
ICT