

# Happie: Project Final Deliverable

---

*the meal planning application using a microservice technology*



FLORIS VOSSEBELD, University of Twente, Netherlands  
EVAN FRANCISZOK, University of Twente, Netherlands

# Introduction

---

Welcome to our detailed report on the ongoing development of our Intelligent Recipe and Meal Planning Application. This document provides an in-depth look into how we've built our application using microservices architecture.

## Project Description

Our project revolves around creating a smart meal planning tool. The goal is to make it easier for users to plan meals, manage their pantry, and create optimized shopping lists that minimize the cost of the groceries. We also give the users the possibility of assigning preferences and diet restrictions. We realised this by using a microservice architecture. Services included: user profile management, recipe suggestions, inventory tracking, and shopping list optimization.

## Report Structure

This report is organized into several main chapters, each focusing on a specific aspect of our project:

- **Motivation:** We explain why we have made a meal planning application and what we aimed to achieve.
- **Business Process:** Here, we explain the main functions that our application performs and how they benefit potential users.
- **Architecture:** This chapter breaks down how we designed our application using microservices architecture.
- **Design Decisions:** We talk about the technology and architecture choices we made and why we made them.
- **Validation:** We explain how we tested our application to make sure it works well.
- **Relevant Information:** Finally, we cover any other important details about our project.

# Table of contents

---

1. <a href="#">Motivation</a>	3
2. <a href="#">Business process</a>	4
3. <a href="#">Architecture</a>	5
4. <a href="#">Design decisions</a>	6
5. <a href="#">Validation</a>	8
6. <a href="#">Relevant information</a>	9
7. <a href="#">Conclusion</a>	10
8. <a href="#">Acknowledgements</a>	10

# Motivation

---

Our motivation for this project has come forward as we both are students that like to sport a lot and don't want to spend a lot of time having to think about what, how and with what ingredients we need to cook. Added to this, one of us does have a food restriction (vegetarian) that makes finding good recipes harder. As both of us are students that don't have a large budget to live with. We would like to minimize the cost of the meals that we will have to make, certainly as we live with other people and we will need to make meals for multiple people.

As normally meal planning can be a time-consuming task that requires a lot of comparisons, searching in books or on the internet. Especially when you want to cook a new meal with some dietary restrictions. Because there is no fun in cooking and eating the same meal over and over again. This resulted in our idea to make a meal planning application that will help with these problems and streamline the experience of cooking.

Our goal is to develop an Intelligent Recipe and Meal Planning Application that addresses these pain points and helps users to make informed decisions that align with their preferences, dietary constraints, while being possible with their lifestyle.

Our objectives include:

- A personalized recommendations page that will provide users with personalized recipe suggestions based on their dietary preferences, diets restrictions, allergies and cost of the meals.
- Tracking of the users Inventory. This enables users to efficiently manage their pantry inventory by tracking ingredient quantities and expiration dates
- Generating optimized Shopping Lists that take into account existing pantry inventory, planned meals, budgetary constraints, and supermarket promotions.
- Simplified Meal Planning suggestions in a user-friendly interface. This interface shows recipe suggestions for the coming days and gives the user the possibility to change suggested meals if they are not wanted.
- These features should be implemented in a flexible and scalable designed application that allows ease of integration with external services, future improvements and expansions.

With these objectives we will try to create an application that will solve our problems with meal planning, while being a good submission for the SOA course.

# Business process

---

- The business process(es) that are supported by your solution.

# Architecture

---

- The architecture of your solution, which should be properly explained and motivated. In the architecture, you should describe the services that you defined and how they work together to solve the problem. Here you should also justify your choices of synchronous and asynchronous services, and message queues.

Our Meal Planning Application is designed with a microservices architecture, this ensures scalability, flexibility, and modularity. because we use this architecture we are able to intergrate various services, each responsible for specific functionalities. In this section, we will go over the architecture of our solution, describing the services defined and how they collaborate with each other.

In our application we implemented the following microservices:

- *Meal planning User Interface*: Eventhough we do not call it a service, it is. This service provides the user with the UI and represents the data from the other services so the user can easily view and interact with the data.
- *Meal Planning Service*: Facilitates meal planning by providing the best fitting personalized recipe suggestions, taking into account the current inventory availability.
- *Recipe Suggestion Service*: This service provides the Meal Planning Services with the meal suggestions based on the user preferences.
- *Inventory Management Service*: Tracks pantry items and their quantities, ensuring accurate inventory management and helps to facilitate informed meal planning and shopping list generation.
- *Shopping List Optimization*: Manages the generation and optimization of shopping lists based on user meal plans, pantry inventory, and current supermarket prices for the items.
- *User Profile Service*: Responsible for managing user profiles, including storing and retrieving user preferences, dietary restrictions, and allergies.
- *Jumbo Price Service*: Retrieves price information from the Jumbo supermarket, enabling cost-effective shopping list generation.
- *Albert Heijn Price Service*: Like the Jumbo Price Service, this service get the price information from the Albert Heijn supermarket. Getting prices of multiple supermarkets makes price comparison possible.

## Architecture Explanation:

Our solution comprises several interconnected components, each fulfilling a specific role in the meal planning process. The meal planning user interface serves a bit as the central hub, orchestrating communication between a few services that in turn will call other services. This User interface provides a user interface for interaction.

## Synchronous vs. Asynchronous Services:

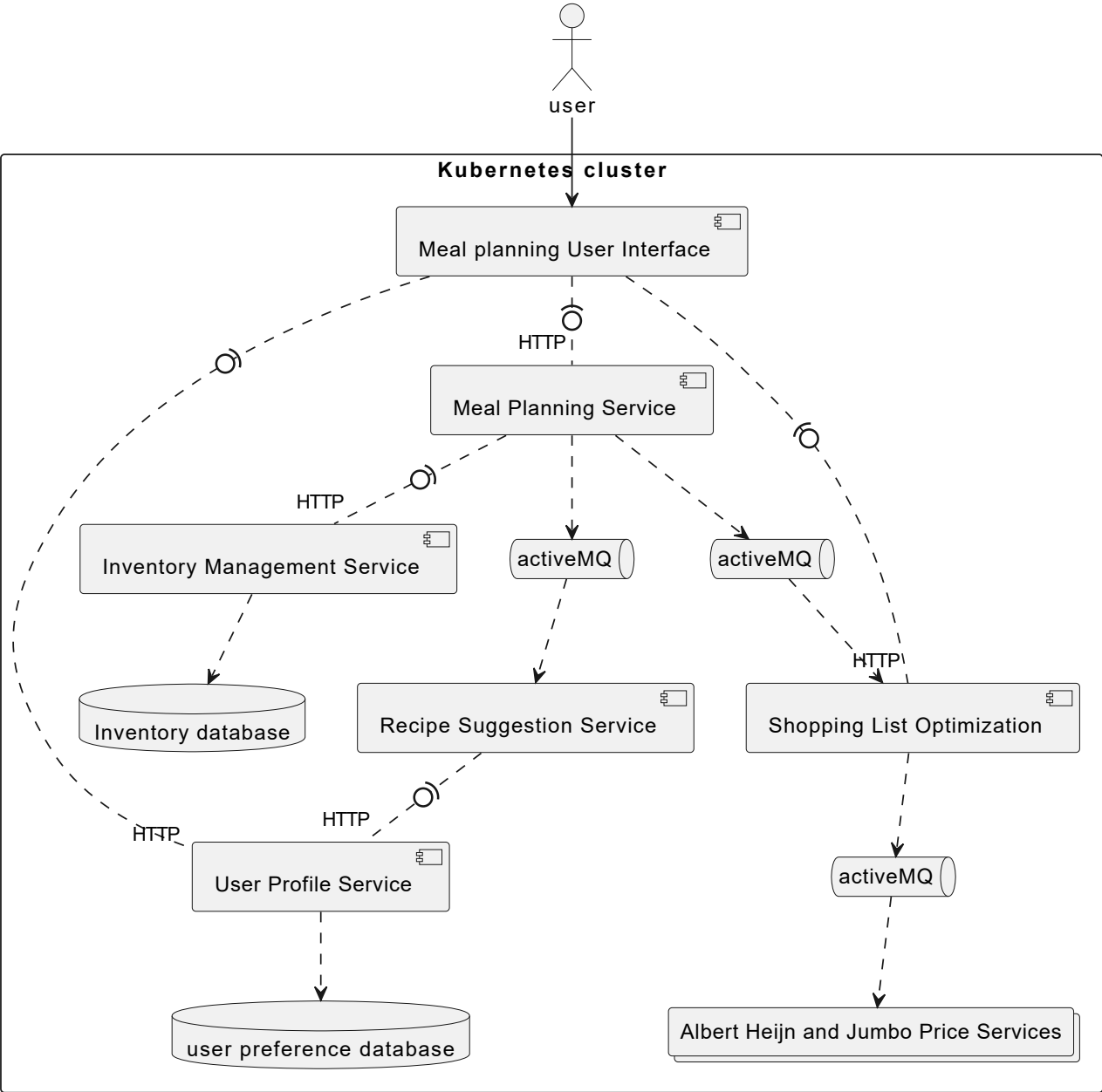
- **Synchronous Communication**: We opted for synchronous communication between the user interface and the Profile, List and Planning services. This choice was motivated by the low computational overhead of these interactions and the need for real-time data retrieval. By utilizing synchronous communication via simple HTTP requests, we ensure efficient and responsive user interactions.

- **Asynchronous Communication:** In contrast, we employ asynchronous communication between the List Service and the Supermarket Service, as well as between the Planning Service and the Inventory and List services. This decision was driven by the desire to avoid blocking the application while fetching price information from third-party APIs and to streamline the meal planning process. By utilizing message queues, we can submit price inquiries in bulk and continue processing other tasks while awaiting responses, thereby enhancing the overall efficiency and responsiveness of our application.

#### **Justification of Message Queues:**

Message queues play a crucial role in facilitating asynchronous communication between services, particularly in scenarios where there is a need to decouple components and handle varying processing times. For instance, the List Service asynchronously queries the Supermarket Service for price information using message queues, allowing for seamless integration with multiple supermarkets and efficient shopping list optimization.

In summary, our microservices architecture enables the modular design and seamless integration of various components, facilitating efficient communication and collaboration to solve the meal planning problem. By leveraging both synchronous and asynchronous services, along with message queues, we ensure scalability, flexibility, and responsiveness in our solution, ultimately enhancing the user experience and promoting healthier eating habits.



# Design decisions

---

- The more concrete/specific design decisions, by explaining why you did (and didn't ) use different technologies such as SOAP, REST and Message queue. Consider giving a summary table listing the services with the information about why you used (or did not use) these technologies and communication styles.

In the development of our Intelligent Recipe and Meal Planning Application, we made several concrete design decisions aimed at optimizing performance, scalability, and user experience. Each decision was carefully considered based on the specific requirements and characteristics of our application.

## Synchronous Communication

One of our design decisions was to employ synchronous communication between the Meal Planning Service and the Inventory Management Service, as well as between all services and the User Profile Service. This choice was made due to the relatively low computational overhead of these interactions, as both services are operated internally and do not require extensive processing. By utilizing simple HTTP requests, we ensure straightforward and efficient communication between these components.

## Asynchronous Communication with Message Queues

In contrast, we opted for asynchronous communication between the Shopping List Optimization Service and the Supermarket Services using message queues. This decision was motivated by the need to avoid blocking our application while fetching prices from third-party APIs, which could introduce delays and hinder responsiveness. By employing message queues, we can submit price inquiries in bulk and continue processing other tasks while awaiting responses. This approach enhances the overall efficiency and responsiveness of our application.

Currently, our implementation involves each supermarket having its own message queue, with a dedicated supermarket service listening to incoming price inquiries. Upon receiving a request, the service processes it and sends back a response containing the product ID, price, and store name, facilitating seamless integration with multiple supermarkets.

## Asynchronous Communication with Websockets

Additionally, we chose to implement asynchronous communication between the Meal Planning Application and the Meal Planning Service using websockets. This decision was driven by the need for real-time updates and a responsive user interface, particularly during the generation of recipe suggestions, which may involve waiting periods. By utilizing websockets, we ensure that results are delivered promptly to the client, even in scenarios where processing times vary.

Our current implementation includes an endpoint for retrieving user preferences via HTTP GET requests and a user interface for viewing and editing preferences. Similarly, a websocket connection facilitates communication between the Meal Planning Application and the Meal Planning Service, allowing for seamless interaction and real-time updates.

## Summary Table of Design Decisions



Service	Communication Style	Reasoning
Meal Planning & Inventory Management	Synchronous (HTTP)	Low computational overhead; Internal operations
All Services & User Profile Service	Synchronous (HTTP)	Internal operations; Querying data from own database
Shopping List Optimization & Supermarket Services	Asynchronous (Message Queue)	Avoid blocking application; Fetch prices from third-party APIs
Meal Planning Application & Meal Planning Service	Asynchronous (Websockets)	Real-time updates; Responsive user interface; Prompt delivery of results, even during processing

These design decisions were made with careful consideration of our application's requirements and objectives, aiming to optimize performance, enhance user experience, and facilitate seamless integration of services. As our implementation progresses, we remain open to refining these decisions based on evolving needs and feedback from users and stakeholders.

# Validation

---

- Concrete evidence that you have validated your system, with testing and usage information.

## Relevant information

---

- Any other relevant information/knowledge that is necessary to appreciate your efforts in this project. The report should be complete and detailed enough so that the teachers don't need to look into the code to understand what has been done.

# Conclusion

---

In conclusion, the development of our Intelligent Recipe and Meal Planning Application represents a significant step forward in addressing the challenges associated with traditional meal planning methods. Through the implementation of a microservices architecture and thoughtful design decisions, we have created a versatile and user-centric solution that offers personalized recipe suggestions, efficient inventory management, and optimized shopping lists.

Throughout the project, we remained committed to our objectives of streamlining the meal planning process, enhancing user satisfaction, and promoting healthier eating habits. By leveraging advanced technologies such as websockets, message queues, and RESTful APIs, we have achieved a high level of performance, scalability, and responsiveness in our application.

Our validation processes, including rigorous testing and user feedback, have helped ensure the reliability, usability, and effectiveness of our solution. We have successfully demonstrated the feasibility and viability of our architecture, while also considering regulatory requirements such as GDPR compliance.

Looking ahead, there are opportunities for further enhancement and refinement of our application. Future iterations may involve expanding the range of services offered, integrating additional features such as nutritional analysis or meal sharing capabilities, and optimizing performance to accommodate growing user demand.

Overall, the Intelligent Recipe and Meal Planning Application represents a significant achievement in leveraging technology to simplify and enhance the culinary experience for users. We are confident that our solution will continue to make a positive impact in the lives of users, helping them make more informed decisions about their meals and ultimately leading to healthier, more sustainable eating habits.

# Acknowledgements

---

In this project we have used ChatGPT 3.5 and 4.0 as a supportive tool, providing structure to our paper and aiding in the identification and correction of spelling mistakes. We did also use ChatGPT for debugging and understanding errors while we were developing the application.

It is important to note that all content generated by ChatGPT underwent thorough review and validation by us. We take full responsibility for the final output.