NutriScan – Nutrition Analysis System

Hugo Ribeiro[1], Filipe Soares[2], Gonçalo Mendes[3], João Serra[4], Mariana Neves[5] and Tiago Gonçalves[6]

Instituto Politécnico do Porto, Porto, Portugal
[1]1141033@isep.ipp.pt, [2]1220823@isep.ipp.pt,
[3]1250421@isep.ipp.pt,[4]1201350@isep.ipp.pt,
[5]1211264@isep.ipp.pt,[6]1140331@isep.ipp.pt

Abstract. Growing public awareness of the connection between diet and health has increased the need for accessible and comprehensible nutritional information. To address this, we developed NutriScan, a web-based expert system designed to provide real-time nutritional analysis of food products. The system integrates barcode recognition with a knowledge base powered by Drools and Prolog inference engines, enabling intelligent reasoning over nutritional data. NutriScan offers detailed product evaluations and scoring. Through personalized user profiles, it identifies potential allergens, generates tailored alerts, and suggests healthier alternatives. The architecture combines both Java (Drools) and Prolog inference back-end components to explore the impact of two different technologies over AI techniques. The system demonstrates the integration of symbolic AI through Prolog-based logical inference and a structured knowledge database, showcasing how expert systems can deliver transparent, rule-driven nutritional analysis and decision support.

While both Drools and Prolog can be applied to rule-based reasoning, their underlying mechanisms differ substantially: Prolog employs backward chaining for logic-based inference, facilitating complex reasoning and knowledge representation, whereas Drools applies forward chaining to enable efficient, scalable rule evaluation with greater implementation clarity.

Overall, NutriScan leverages expert system principles and AI reasoning to support informed and health-conscious consumer decisions. The successful development and validation of NutriScan highlight the effectiveness of combining distinct inference paradigms to create intelligent, user-oriented decision-support tools.

Keywords: AI software, Expert systems, Knowledge Base, Drools, Prolog

1 Introduction

In recent years, growing public awareness of the relationship between diet and health has highlighted the importance of providing consumers with accessible and reliable nutritional information [19]. Poor nutrition and unhealthy diets are major contributors to rising global rates of obesity, diabetes, and heart disease, highlighting the need for better nutritional awareness [23] and decision-making tools. Despite the increasing availability of food labeling systems, many individuals still struggle to interpret complex nutritional data and evaluate the health implications of their food choices [8]. This challenge has created a demand for intelligent, user-oriented tools capable of simplifying the process of understanding and comparing food products.

To address this need, NutriScan was developed – a web-based expert system that performs real-time nutritional analysis of food items. The application integrates barcode recognition with artificial intelligence reasoning mechanisms implemented using Drools and Prolog inference engines. By combining a structured knowledge base with symbolic AI techniques, NutriScan provides accurate and personalized evaluations, as well as alerts tailored to each user's dietary profile and restrictions.

This project aims, not only to create a functional application that supports consumers to have real-time information about the nutrition, but also to apply AI techniques over software engineering, to use different programming languages for implementing an AI solution and to measure the impact of different technologies.

2 State of the Art

Artificial Intelligence (AI) is a branch of computer science, that uses algorithms, data and hardware to simulate human intelligence, make decisions, discover patterns or perform some sort of action [10]. Currently, it is composed by the following AI techniques [15]: Machine Learning, Deep Learning [10], Natural Language Processing [18], Expert Systems [14] Computer Vision, Robotics [18], Reinforcement Learning [1].

Each subject focuses on a specific dimension of intelligence, such as perception in computer vision [18], reasoning in expert systems [14], or adaptation in reinforcement learning [1] and their integration within software applications allows for the development of highly autonomous and context-aware solutions.

2.1 Expert Systems

Expert systems are a branch of Artificial Intelligence, that can imitate and reproduce the flow of reasoning and decision making executed in human experts' brains to derive optimal solutions [14]. These systems, consists in a knowledge base, inference engine and user interface [2].

Knowledge bases are structured repositories designed to store, manage and retrieve information, enabling systems to solve complex problems through reasoning and inference [20]. Information is present in the knowledge base using facts and rules. **Facts** are represented by basic, unambiguous pieces of information considered true within a specific domain [9], such as "Window is open" or "Room is occupied". **Rules** are the are conditional statements that define relationships between facts and guide the system's reasoning process [9]. As an example, "IF window is open and room is not occupied THEN close the window".

The **inference engine** applies logical reasoning to the knowledge base to derive conclusions, distinguishing expert systems from other computer systems [2]. The **explanation module** provides users with justifications or reasoning paths that led to a certain conclusion, increasing transparency and trust in the system's decisions [6]. The **user interface** allows interaction between the user and the system, enabling input of data and interpretation of outputs in an intuitive way [7].

By using the knowledge and validation of expert humans on a given subject, it becomes easier to achieve more accurate results and derive optimal solutions [14] within AI systems. It can be used in areas like medicine, business, computer science, law, defense, education, mathematics, engineering and more [3].

2.2 Programming Languages

AI techniques can be applied through various resources, especially programming languages. Languages such as Prolog, Java, and Python can be used to develop intelligent systems capable of reasoning, learning, and problem-solving [13].

Prolog is a declarative language, that allows facts and rules to be represented uniformly, making it useful for rule-based expert systems. Its goal-driven inference uses depth-first search with backtracking, and features like uncertainty handling or explanations can be added by extending predicates. Expert systems can be fully implemented in Prolog without a host language, as demonstrated by both small and large-scale applications [14].

However, Prolog has limitations: its rigid control strategies can make some inferences inefficient, and procedural constructs, like "assert" or "cut", can reduce program clarity, by making it less declarative. Knowledge base structuring is limited, and mixing domain and control knowledge can reduce modularity, complicating system extensions such as explanation mechanisms [14].

Java is an object-oriented programming language, that is widely used for building applications. Its fusion with artificial intelligence offers a powerful synergy, enabling developers to create intelligent systems and applications with efficiency, robustness and scalability [21].

Java provides many AI libraries and tools, offering many solutions for programmers to implement AI algorithms and techniques [21]. Among other tools, there is available **Drools**, a Business Rules Management System (BRMS) and a rules engine [5]. It is a rule-based approach that allows to implement an Expert System, with a knowledge base holding the rules; internal types, and processes, representing the system's knowledge; the working memory stores the facts; and these facts against the rules that infers the facts [21].

By the other hand, Java shows to have runtime overhead and performance concerns, and it tends to have more complexity and verbosity when comparing to other languages, like Python [13].

2.3 Nutrition Apps

The advances in information technology have made nutrition applications more accessible and widely used. Artificial intelligence enhances these apps by enabling features like automated food recognition and personalized dietary recommendations, making diet monitoring more convenient and tailored to individual needs [22]. This personalization aims to optimize nutrition plans, potentially improving health outcomes for users [22].

Nutrition applications are described to be a positive asset for consumers, having the potential to allow them to have more informed decisions regarding their food intake [12] and support disease treatment and control, like diabetes [16]. Some applications were confirmed to be a major improvement in monitoring diet quality and weight control and, preventing the development of diet associated chronic diseases [12].

Nevertheless, there are still several challenges that must be addressed, such as the lack of standardization of food databases and the difficulty in accurately recognizing foods and measuring nutrient content [16].

3 Project

To address the need of consumers for nutritional advising, it was created NutriScan, a web application, that provides an intelligent system to allow consumers to evaluate in real-time food products and its nutritional value, having into account their profile and food restrictions.

NutriScan offers a **smart scanning** supported by mobile devices, with automatic recognition of barcodes, integration with standard product databases and offline processing of previously searched products. With **consumer profiles**, the application considers the restriction management, of allergies, intolerances, and dietary preferences; provides critical alerts when detecting hazardous ingredients; holds contextual analysis. Its **analysis system**, provides a nutritional analysis, taking sugars, salt, fats, additives and nutritional value in consideration. It identifies food properties like vegan, organic, and gluten-free products. It used visual indicators, as traffic light color system and pictograms for quick user interpretation and alarm.

To sustain the previous functional requirements, it was defined the following technical goals for this project:

- Implement a robust barcode scanning system.

- Develop a Drools/Prolog rules engine for intelligent product classification.
- Implement a personalized profile system with dietary restriction management.

3.1 Application Architecture

The architecture of NutriScan project is composed by the following components:

- **Java backend server** – is a Java 1.8 API server application that handles requests from UI, requests to Prolog and Drools and database connections.
- **Prolog backend server** – is a Prolog API server application that handles Prolog logical inference system.
- **Drools backend server** – is a Java application that uses Drools to handle rules engine.
- **UI application** – is the user interface from the application, written in React, with the main feature allowing end users to interact with out internal back-end APIs.

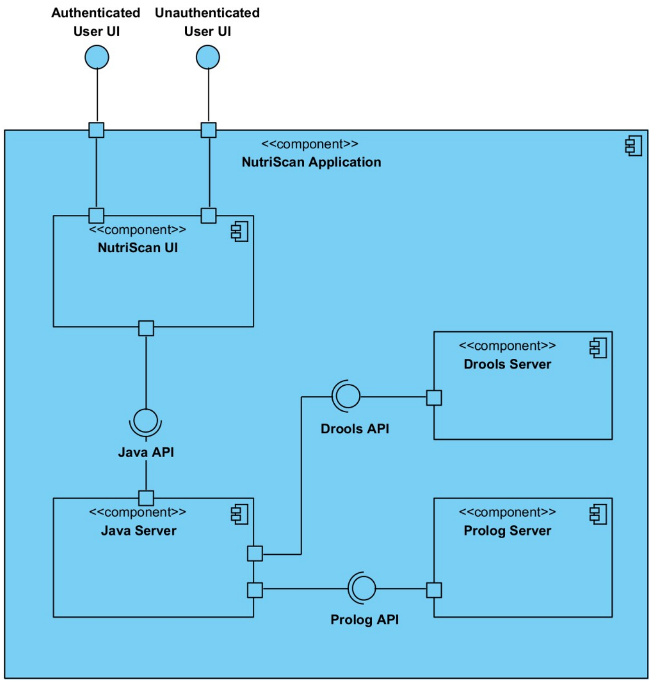

Fig. 2. Level 2 Logical View Component Diagram.

Using this architecture, we can divide the application into different components according to their technological requirements. The front-end is naturally separated due to its UI nature and React stack. A main Java server responsible for managing communication between the UI and the various back-end components. With this AI-oriented approach, we are able to maintain two distinct models, which is an essential aspect of this project's goal, to study and compare the results of Java and Prolog implementations.

By using C4 with 4+1 model, NutriScan application architecture could be represented. C4 model, divides software architecture into different levels: context (level 1); containers (level 2); components (level 3); and code (level 4) [4]. 4+1 model divides software in logical view, process view, development view, physical view and uses cases [11].

3.2 Expert System

For building NutriScan it was used an expert system, to ensure that the knowledge base reflects accurate, context-specific expertise, allowing the system to simulate human-like judgment more effectively.

Interacting with experts is essential for a rule engine, as they provide the base knowledge that shapes its facts, rules, and inference. Interviews were run with three experts from different backgrounds (marketing and nutrition).

Beatriz Almeida and Flávio Gart, are two mentors with experience as directors in marketing agencies, support the NutriScan project by validating the application features, advising about project's potential for successful market adoption and user engagement, validating the application features and business decisions.

Daniela de Sousa, a nutritionist specializing in Clinical and Sports Nutrition, provides essential nutritional expertise to NutriScan. She helps shape the app's health recommendations and dietary analysis features, by providing the nutrition information and the rules to infer with the existing knowledge base.

3.3 Rule Implementation

After having the meetings and discussions with the experts, it was defined two major groups of rules to implement in Drools and Prolog, as it follows:

Core Rules: General Nutritional Assessment. These rules evaluate products based solely on their nutritional composition and ingredient profile, independently of the user's health data. They identify general positive and negative nutritional attributes, such as: High or low content of salt, sugar, fat (total/saturated), and energy density; Positive nutritional features such as low fat, low sugar, high fiber, or protein source. Each rule corresponds to a clear nutritional threshold derived from expert guidelines.

Personalized Alerts: User-Specific Warnings. These rules integrate user health profiles (e.g., hypertension, diabetes, pregnancy, childhood) to generate personalized or critical alerts. Examples include: Alerts for high salt, sugar, or fat intake in users with related conditions; Critical warnings for allergens such as gluten; Specific advisories for children (e.g., artificial colorants, added sugars), pregnant women (artificial sweeteners), and sensitive groups (polyols or low energy foods).

4 Results and Discussion

This section presents the results obtained from the development and evaluation of the NutriScan application.

4.1 Application

The built application offers a clean and modern view for users, with the proposed features implemented. It allows users to create a profile with detailed information about their food restrictions, properly scanning a product barcode, select different analysis tools and give user food suggestions and alerts.

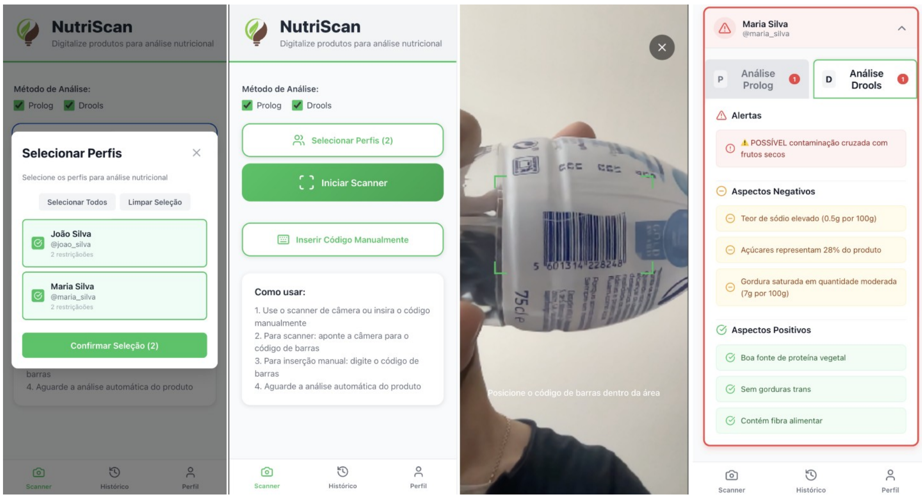

Fig. 3. NutriScan user interface and its features. From left to right: profile selection; analysis and scanning menu; barcode scanning view; analysis results view.

Login. Is the homepage of the NutriScan that allows users to authenticate on the website. Only visible for non authenticated users.

Account Creation. Allows non registered users to create an account. Only visible for non authenticated users.

Create and Edit Profile. This view allows users to create and edit profiles, to fill their name, gender, birth date, if it is pregnant or breastfeeding. It is in this section that users insert information about is allergies, diseases like hypertension, diabetes or celiac disease.

Profile Management. Allows users to manage multiple profiles from the same account, with options to list, create, see details, remove and update profile options.

Analysis Selection. This view holds the feature of selecting analysis methods (Prolog and/or Drools), selecting profiles for analysis, a button for scanning using the camera or the option to insert manually the bar-codes.

Barcode scanning. It shows the camera view, in order to start finding and scanning a code, from a product packaging.

Product analysis result. This view shows details about the scanned product (if it is found) and displays its description, nutritional value and the profile analysis for each one of the profiles. There is a tab separating Prolog and Drools, and inside of each one, it shows the alerts for the found food ingredient restrictions and allergies, positive and negative aspects according to nutritional value.

Historic. Allows to consult previous analysis done by the users.

4.2 Test Case

To validate the application, a complete acceptance test was conducted, covering all features from user registration to the history of analysis results, following a *happy path* scenario.

1. A user account was registered by entering an email address and password, after which authentication was successfully completed.
2. Within the NutriScan application, a consumer profile was defined representing a three-month-old female with barley allergy.
3. The analysis parameters were then configured, by selecting the previously created profile, both the Drools and Prolog inference models, and a product was scanned using its barcode.
4. The scanned product corresponded to a chocolate cereal, and the system displayed its nutritional information per 100g: 385 kcal, 24.8g of sugar, 0.2g of salt, and 4.6g of fat.
5. The analysis produced the following results in both inference models:
    - **Alerts:** not recommended for children under 12 months due to added sugars; contains barley, which is contraindicated when allergic to this ingredient.
    - **Negative aspects:** high sugar content.

- **Positive aspects:** high fiber content; low salt; low fat.

6. Finally, it was verified that this analysis was correctly stored in the user's history list.

These results confirm that both inference engines, Drools and Prolog, generated consistent outputs, and that the system's core functionalities (registration, analysis, and data persistence) operated as expected.

5 Conclusion

This study presented the development of a nutrition-focused application that integrates an expert system using Prolog for logical inference and Drools for rule-based decision management.

The application was successfully developed to provide personalized nutritional guidance based on individual user profiles. The results demonstrate the potential of expert systems, to enhance decision support and validate specialist knowledge in nutrition-related applications.

Regarding Drools and Prolog implementations, they can be used for similar goals (rule-based reasoning or expert systems), but their underlying nature and reasoning processes are fundamentally different. Prolog is a logic programming language that uses backward chaining to infer conclusions from a set of facts and logical rules, making it well suited for complex reasoning and knowledge representation. In contrast, Drools is a forward-chaining rule engine designed for efficient rule evaluation. While Prolog emphasizes declarative logic and inference, Drools focuses on scalable rule execution and its implementation seems to be have more readability.

5.1 Limitations and Future Work

The current knowledge base is constrained by the existing scope, which may not cover all nutritional scenarios or individual variations. Additionally, the system's recommendations depend on the quality of user-provided data and have not yet been clinically validated. Future work should focus on expanding the rule set and conducting broader evaluations with expert and user feedback.

References

1. Aggarwal, C., Reinforcement learning. In: *Artificial Intelligence*, pp. 343–384. Springer, Cham (2021).
2. Asemi, A.: Intelligent libraries: a review on expert systems, artificial intelligence, and robot. Library Hi Tech 39(2), 412–434 (2021).
3. Bavakutty, M., Salih, M.: Research on Library Computerisation. 1st edn. Ess Ess Publications (2006)
4. C4 Model, https://c4model.com, last accessed 2025/10/08

5. Drools Expert User Guide, Chapter 1: The Rule Engine, https://docs.drools.org/5.2.0.M2/drools-expert-docs/html/ch01.html, last accessed: 2025/10/25.
6. Expert Systems: Definition, Components, and Applications, Studylib. https://studylib.net/doc/25943845/ai-unit-5, last accessed 2025/10/28.
7. Füßl, A. and Nissen, V. "Design of explanation user interfaces for interactive machine learning using the example of a knowledge graph-based approach to explainable process analysis," Requirements Engineering, vol. 30, no. 1, pp. 81–108 (2025).
8. Grunert, K., Wills, J.: *A Review of European Research on Consumer Response to Nutrition Information on Food Labels*. Journal of Public Health, 15(5), 385–399 (2007).
9. Gupta, I., Nagpal, G.: Chapter 5: Expert Systems. In: *Artificial Intelligence and Expert Systems*, pp. 71–94. Mercury Learning and Information, Berlin/Boston (2020).
10. Janiesch, C., Zschech, P., Heinrich, K.: Machine learning and deep learning. *Electronic Markets* 31, 685–695 (2021).
11. Kruchten, P.: Architectural Blueprints—The "4+1" View Model of Software Architecture. *IEEE Software* 12(6), 42–50 (1995).
12. Kong, N., Moy, F., Ong, S., Tahir, G., Loo, C.: MyDietCam: Development and usability study of a food recognition integrated dietary monitoring smartphone application. Digit. Health 9, 1-18 (2023).
13. Marini, D., Khamlichi, A., El Alaoui, I.: Comparative Analysis of Four Programming Languages for Machine Learning. Int. J. Information and Systems Sciences 30(6), 1069–1077 (2023).
14. Matsuzaka, Y., Yashiro, R.: AI-Based Computer Vision Techniques and Expert Systems. AI 4, 289–302 (2023).
15. Mukhamediev, R., Popova, Y., Kuchin, Y., Zaitseva, E., Kalimoldayev, A., Symagulov, A., Levashenko, V., Abdoldina, F., Gopejenko, V., Yakunin, K., Muhamedijeva, E., Yelis, M.: *Review of Artificial Intelligence and Machine Learning Technologies: Classification, Restrictions, Opportunities and Challenges.* Mathematics, 10(15), 2552 (2022).
16. Rahman, L., Papathanail, I., Brigato, L., Spanakis, E., Mougiakakou, S.: *Food Recognition and Nutritional Apps*. (2023).
17. Rossi, G.: *Uses of Prolog in Implementation of Expert Systems*. *New Generation Computing*, 4(3), 321–329 (1986).
18. Ruiz, P., Fusco, J., Glossary of Artificial Intelligence Terms for Educators. Digital Promise (2024). https://digitalpromise.org/wp-content/uploads/2024/12/CIRCLS-Glossary-of-Artificial-Intelligence-Terms-for-Educators.pdf, last accessed 2025/10/28.
19. Silva, P., Araújo, R., Lopes, F., and Ray, S., Nutrition and Food Literacy: Framing the Challenges to Health Communication. Nutrients 15(22), 4708 (2023).
20. Sriram, R., Knowledge-Based Expert Systems: An Overview. In: *Intelligent Systems for Engineering*, pp. 1–22. Springer, London (1997).
21. Vyas, B.: Java-Powered AI: Implementing Intelligent Systems with Code. Journal of Science & Technology 4(6), 1–12 (2023).
22. Wang, X., Sun, Z., Xue, H., An, R.: Artificial Intelligence Applications to Personalized Dietary Recommendations: A Systematic Review. *Healthcare (Basel)* 13(12), 1417 (2025).
23. World Cancer Research Fund International and NCD Alliance: *The Link Between Food, Nutrition, Diet and Non-Communicable Diseases*. World Cancer Research Fund International, London (2014). Available at: https://www.wcrf.org/research-policy/library/link-between-food-nutrition-diet-ncds/