A Technological Proposal for Optimizing Order Handling in Culinary Spaces of Mall Plaza

Daniel Giraldo Sanclemente[1],

[1] Departamento de Computação e Sistemas
Universidade Federal de Ouro Preto – João Monlevade, MG– Brazil

daniel.sanclemente@aluno.ufop.edu.br

Abstract. *The developed software offers a solution for shopping malls where people with different culinary tastes can place orders from various restaurants, allowing each restaurant to manage its orders. It utilizes a MongoDB database, as well as Python and Flask for backend development, and JavaScript and React for frontend development..*

1. Introduction

Food courts located within shopping malls or supermarkets are restrictive with their orders, as well as with their management systems. Each customer must go to the restaurant they desire and place their respective order; however, there are times when large groups of people arrive at these places, and not all of them share the same culinary interests. As a result, they divide into groups according to the restaurant and place their respective orders before regrouping. Thus, the idea behind this technological solution is to offer customers the possibility of making a single order composed of sub-orders, where each sub-order reaches the involved restaurant for it to carry out production and dispatch the order to the customer. This allows people to place orders at different restaurants from the table they select.

One of the key aspects of this solution is the chosen database, which in this case is of the NoSQL type. This term first emerged in 1998 by Carlo Strozzi to name his small relational database management system that did not use SQL for data manipulation (Strozzi, 1995). According to Haseeb and Pattun (2017), these types of systems are particularly suitable for flexible architectures, a crucial characteristic for this system. The required data structure includes orders that contain sub-orders, which in turn must contain items. This underscores the importance of having a data manager that not only allows for efficient handling of this hierarchy but also offers the necessary flexibility to adapt to changes in the organization of data.

Mapanga and Kadebu (2013) suggest that these database management systems are characterized by flexibility in data storage, and one of their main advantages is performance, due to the very low read and write latencies as they are distributed across various servers. The food courts in supermarkets experience peaks where the amount of

data reading and insertion can be extremely high, as a large number of tables may be making requests from various restaurants. This same author suggests that these systems excel in terms of availability, making it important for the database to be available during peak hours mentioned earlier.

This type of tool is adaptable for both a single shopping mall or supermarket and for multiple establishments. In the case of a chain of shopping malls wishing to implement the system across all its locations, this is entirely feasible. One of the significant strengths of NoSQL databases is their horizontal scalability (Kuznetsov and Poskonin, 2014), allowing for the management of an increasing volume of data without quickly encountering a limit to their capacity.

On the other hand, data analysis has become a technological boom that drives competitiveness. These database management systems allow for the collection of large volumes of information, which shopping malls and restaurants can leverage to optimize their advertising campaigns. Through an efficient big data analysis process, it is possible to generate valuable insights to enhance decision-making (Alotaibi et al., 2019).

2. Methology

For the preparation of the references for this document, a literature review is initially conducted on Google Scholar using the search equation ("nosql" and "applications"), and later a third concept is added ("nosql" and "applications" and "data access"). Based on these equations, the abstracts of each article are read to identify which ones may be most relevant for the research, resulting in the identification of five potential articles that were cited.

On the other hand, the software consists of three parts:

Database: For this need, MongoDB is used, which is a NoSQL database. This tool is chosen due to the hierarchical nature of the data, as this is one of MongoDB's greatest strengths.

Backend: It is developed in the Python programming language using the Flask library, all to create an API that allows the system to function and connect with the database. The architecture of the backend is based on the model-controller pattern (Figure 1).

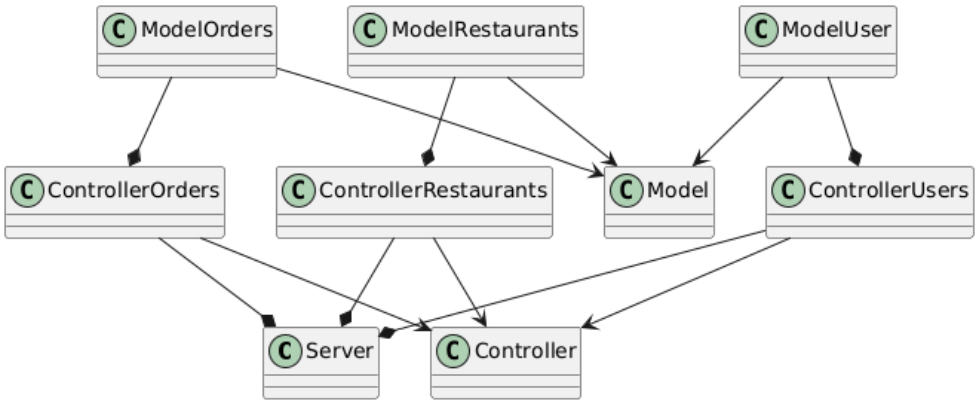

Figure 1. Arquitectura backend

Frontend: The frontend is developed in JavaScript using the React library and react-router-dom. This setup allows for the creation of a graphical interface that enables users to insert and read data.

3. Results

As a result, a functional software is obtained that has three main views.

User View: This view, as shown in Figure 2, allows users to see the created orders, as well as everything related to them. Additionally, the user can create new orders starting from this view and add products from each restaurant to the order, visualizing the status of the sub-order..

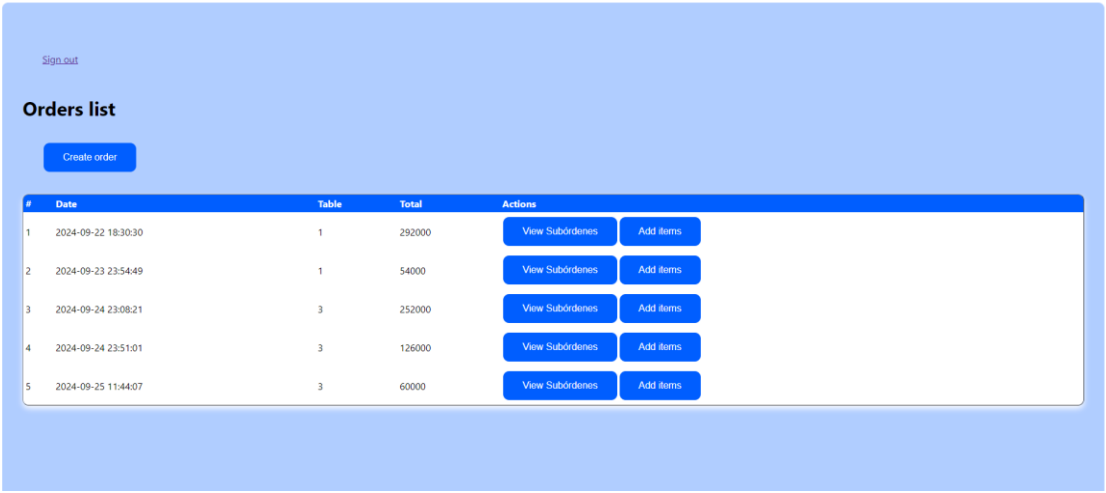

Figure 2. Users view

Restaurant View: From this view, represented in Figure 3, each restaurant can see the orders and the associated sub-orders relevant to it, allowing them to know what they need to produce and deliver to the customer. Similarly, the restaurant will be able to modify the status of the order to indicate that the orders that have already been dispatched are marked as completed.

Figure 3. Restaurants view

Admin view: The person in charge of the system will be able to view all the information related to users and restaurants, as shown in Figure 4.

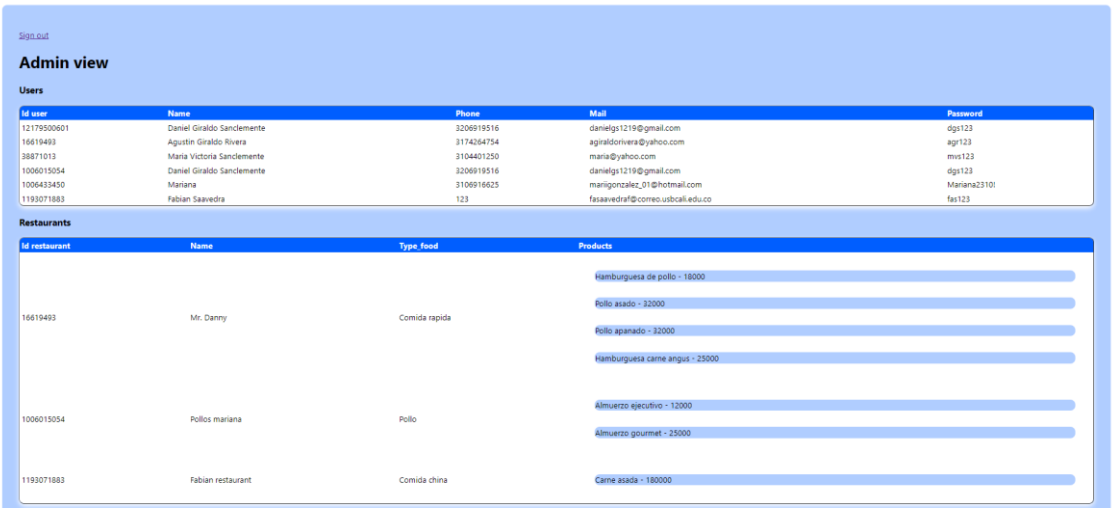

Figure 4. Admin view

4. Discussion

The technological proposal described reflects an innovative solution to enhance the user experience in food courts by allowing the creation of orders intended for different restaurants. This strategy not only addresses the challenge faced by groups with diverse culinary preferences but also optimizes the logistical process by centralizing orders and allowing restaurants to receive only the requests that pertain to them.

However, there are numerous opportunities for improvement that could enhance the software's performance and usability. Improvement opportunities include user experience, new functionalities, better programming practices, among others. The developed software is merely a prototype to validate its approach and functionality.

5. Conclusions

In conclusion, the proposed technological solution for food courts efficiently addresses the current limitations in managing multiple orders, offering a more convenient and centralized experience for users. By allowing the creation of orders composed of sub-orders for different restaurants, both the operational efficiency of establishments and customer satisfaction are improved.

NoSQL databases are a suitable component for problems requiring flexibility, availability, performance, and horizontal scalability, especially in environments where the data structure may be diverse or evolve over time. Their ability to handle unstructured or semi-structured data enables developers to design systems that adapt to the changes required by the environment, unlike SQL databases, which tend to be too restrictive.

Furthermore, NoSQL databases are designed to provide high availability by distributing data across multiple servers, making them ideal for applications that demand quick responses and continuous operation. Performance is also optimized due to their capability to handle large volumes of reading and writing simultaneously, which is essential in applications with high traffic levels or large data volumes, as seen during peak hours in the food courts of shopping malls.

On the other hand, their horizontal scalability, which allows for the addition of extra servers to increase storage and processing capacity, makes them highly efficient in scenarios requiring large-scale growth, such as if a chain of shopping malls wanted to implement this system.

References

ALOTAIBI, Ashwaq A.; ALOTAIBI, Reem M.; HAMZA, Nermin. Access Control Models in NoSQL Databases: An Overview. JKAU: Computer and Information Technology Sciences, v. 8, n. 1, p. 1-9, 1440 A.H. / 2019 A.D.

HASEEB, Abdul; PATTUN, Geeta. A review on NoSQL: Applications and challenges. International Journal of Advanced Research in Computer Science, v. 8, n. 1, p. 203, Jan.-Feb. 2017. Available at: http://www.ijarcs.info.

KUZNETSOV, S.D.; POSKONIN, A.V. Database Management Systems: A NoSQL Analysis. Programming and Computer Software, v. 40, n. 6, p. 323-332, 2014.

MAPANGA, Innocent; KADEBU, Prudence. Database Management Systems: A NoSQL Analysis. International Journal of Modern Communication Technologies & Research (IJMCTR), v. 1, n. 7, p. 12, Sept. 2013.

STROZZI, C. NoSQL: A relational database management system. Available at: http://www.strozzi.it/cgi-bin/CSA/tw7/I/en_US/NoSQL/Home%20Page. Accessed on: Sept. 26, 2024.Dyer, S., Martin, J. and Zulauf, J. (1995) “Motion Capture White

Paper", http://reality.sgi.com/employees/jam_sb/mocap/MoCapWP_v2.0.html, December.