WareSmart Inventory Management System

Diego A. Medina Molina, Gabriel A. Granell Jiménez, Ricardo Reyes Burgos, Roberto C. Marrero Ortiz
Advisor: Wilson Rivera Gallego
Department of Computer Science and Engineering

Computer Science and Engineering
University of Puerto Rico Mayagüez Campus

1 Problem Statement

Currently, inventory management systems are clunky, inconvenient, and at times completely inaccessible to anyone without a business budget.

Business owners have reportedly expressed desire for higher quality ease of use and additional features out of the systems they pay for, let alone a free one.

2 Problem Background

Small businesses have a 20 percent rate of failure within the first year of operation, 30 percent of failure within the second year and by the end of the decade, the failure rate is an alarming 70 percent. A strong positive relationship between the success rate of a small business and the effective management of inventory exists.[1]

Inventory management should be a “set it up once and forget about it” situation but the biggest problem with this type of service is the prohibiting cost and complexity associated with it. For someone like a small business owner, this leads to financial problems.

3 Objectives

WareSmart aims to develop a user-friendly cloud-based inventory system with the following goals in mind:

1. Design a complete inventory tracker that satisfies all user needs detailed in our survey. Server response time must not exceed 200 milliseconds.
2. Increase user satisfaction by 20% within 6 months by implementing user feedback enhancements and re-surveying testers.
3. Reduce system report generation time to less than 30 seconds for 90% of user interactions by optimizing database queries.

4 Technical Approach

System Architecture:

Frontend

- Vite+React hosted on S3 bucket.
- CloudFront for secure access.
- Route 53 for address resolution.

Backend

- Flask API communicating with Aurora database.
- Integration of Carbone.io for reports.
- Utilization of Barcode Lookup API for product data.

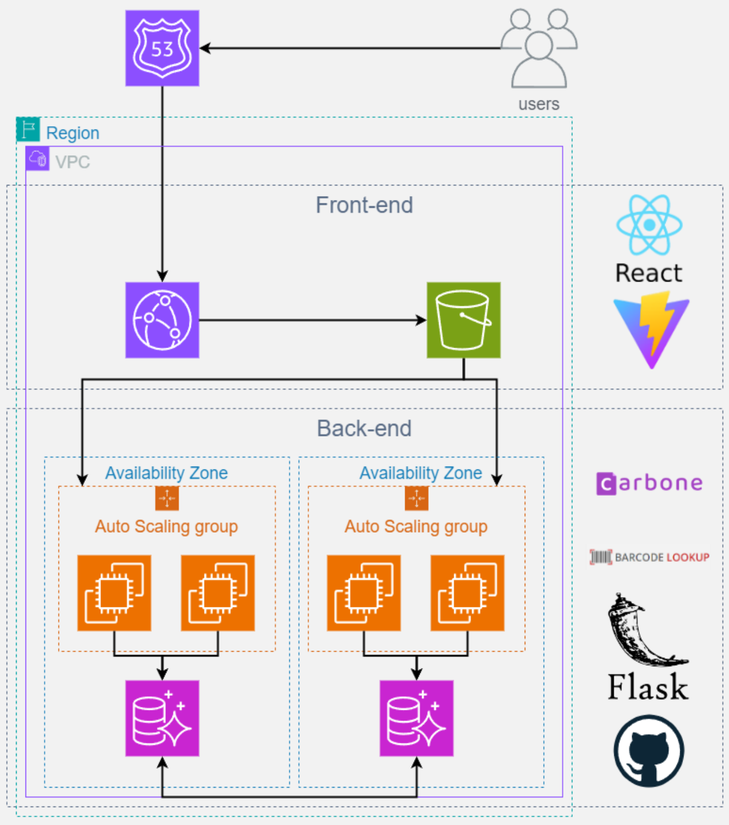

Figure 1. WareSmart System Architecture

User Management

- Business users can manage organization users.
- Roles include manager and employee.

Multi-Branch & Inventory Management

- Manage multiple branches and inventories per branch.

Product Management

- Stock based organization of products for products with different expiry dates, as well as different inventories.

Notifications

- Alerts for low stock onsite and via email.

Recommender System

- ML-based recommendations based on historical sales.

5 Results

The average response time for POST was **127.12 milliseconds** and for GET was **197.14 milliseconds**, suggesting that the system managed to maintain efficiency across operations with no errors. As well as the report generation with an average size **5285.65 bytes** and a consistent response time of **13137.87 milliseconds**.

Table 1. Average Response time for different requests

Type	Name	#Fails	Average (ms)	Min (ms)	Max (ms)	Average size (byte)
GET	//waresmart/product	0	197.14	81	440	1232.56
POST	//waresmart/product	0	127.12	79	330	13.02
POST	//waresmart/reports	0	13137.87	10962	15291	5285.65

Figure 2. Total Request per Seconds and response time

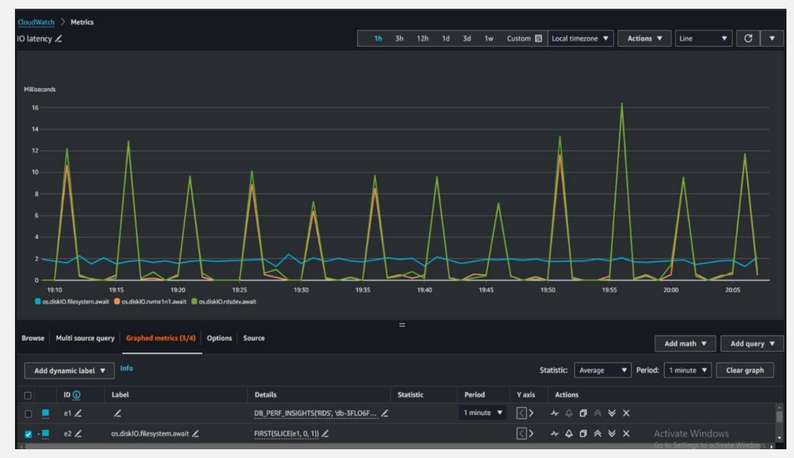

Figure 3. AWS IO Cloud Watch latency Metrics

6 Conclusion

The resulting WareSmart app is a free, easy-to-access inventory management application hosted on AWS where user-survey-requested inventory needs are met.

R References

[1] Medeiro, F. and Perez-Verdu, A. and Rodriguez-Vazquez, A., D. Atnafu and A. Balda, "The impact of inventory management practice on firms’ competitiveness and organizational performance: Empirical evidence from micro and small enterprises in Ethiopia," Cogent Business & Management, vol. 5, 2018.

[2] A. Ihuman, "Deploying a React App using AWS S3 and Cloud Front," Medium, 4 September 2023. [Online]. Available: https://medium.com/@Anita-ihuman/deploying-a-react-app-using-aws-s3-and-cloud-front-c0950808bf03. [Accessed 5 April 2024].

[3] GO-FOR-IT, "Deploying a Flask Application on EC2," Medium, 22 September 2023. [Online]. Available: https://aws.plainenglish.io/deploying-a-flask-application-on-ec2-54cfeb396fa1. [Accessed 5 April 2024].