Reducing Mean Time to Resolution: Developing a Tool to Solve Customer Issues With a Real User Monitoring Service

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

David Hasani

Spring, 2023

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Dr. Briana Morrison, Department of Computer Science

Reducing Mean Time to Resolution: Developing a Tool to Solve Customer Issues With a Real User Monitoring Service

CS4991 Capstone Report, 2023

David Hasani
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
dh8rsv@virginia.edu

Abstract

CloudWatch Real User Monitoring (CW RUM), a service developed and maintained by Amazon Web Services (AWS), has found that when AWS software developers resolve (i.e., debug) customer issues with CW RUM, the experience is often difficult, slow, and overly reliant on assistance from customers. To address this, I developed an application that allows CW RUM team members to quickly, securely, and independently access information about customer accounts and numerous details about the CW RUM service's operational status. I wrote the application in the Python programming language. The application contains twelve different commands, each retrieving and displaying different information, that the user can execute. The major outcome of developing this application was a significant reduction in the average time taken for CW RUM team members to resolve customer issues. The application has thus contributed to markedly improved customer satisfaction, as well as notable simplification of the debugging experience for the CW RUM team. Future work on this application will extend the tool by adding commands not just to display data about the CW RUM service but also to automatically perform various types of analyses of the data so that issue resolution can be further accelerated.

1 Introduction

The promptness and completeness with which customer issues are addressed play a significant role in the long-term performance of any business. With newer products and services, these factors are especially important, as they influence key metrics, such as customer retention rates and the reputation of the firm. In fact, cloud computing providers like AWS often sign agreements guaranteeing certain levels of availability, accuracy, security, and more. Along with these high expectations for a service come high expectations for rapid customer issue resolution. Simply put, when an AWS service stops working or otherwise fails, customers expect the issue to be fixed quickly. There is a ton at stake for them, both financially and reputationally. My work over the summer, developing software which helps engineers quickly access relevant information about the status and health of our service, was thus heavily focused on meeting the high bar set by AWS customers.

2 Related Works

A wide variety of research has been conducted on the importance of customer service for maximizing the financial performance of a business. An analysis of customer expectations management comes from Sheth and Mittal (1996), who note that

when customer expectations are "unjustified, infeasible or unproductive," they need to be "shaped rather than fulfilled" [2]. This commentary challenged my original belief that customer requests should be relentlessly pursued and achieved. Sheth and Banwari posited that customer requests should first be analyzed for feasibility and utility. Realistic expectations need to be set, transparently, from the outset to ensure project success and overall customer satisfaction. Additionally, in a recent interview with Beale (2022), the Chief Executive Officer of the Institute of Customer Service noted that her organization has "been able to prove the link between customer experience, employee engagement and ROI [return on investment]" [1]. Coming from an experienced business professional leading a reputable organization, these comments strongly reinforce the importance of building tools which will help businesses quickly respond to customer needs, optimizing their experience and in turn maximizing the profitability of the business. These points on the importance of customer service also justify the significant time and financial commitment involved with the execution of my summer project.

3 System Design

The software I developed was carefully designed to ensure its requirements would be fully met. The tool, which users interact with through the computer terminal, consists of five primary components. The final design of the tool (code-named "Screwdriver") is shown below. Designing this tool took roughly one week, and my design was reviewed extensively by senior software developers on the CW RUM team. After iteratively working on the design by addressing their comments and re-submitting for further review, I arrived at the design shown in Figure 1 and proceeded with the implementation of the tool.

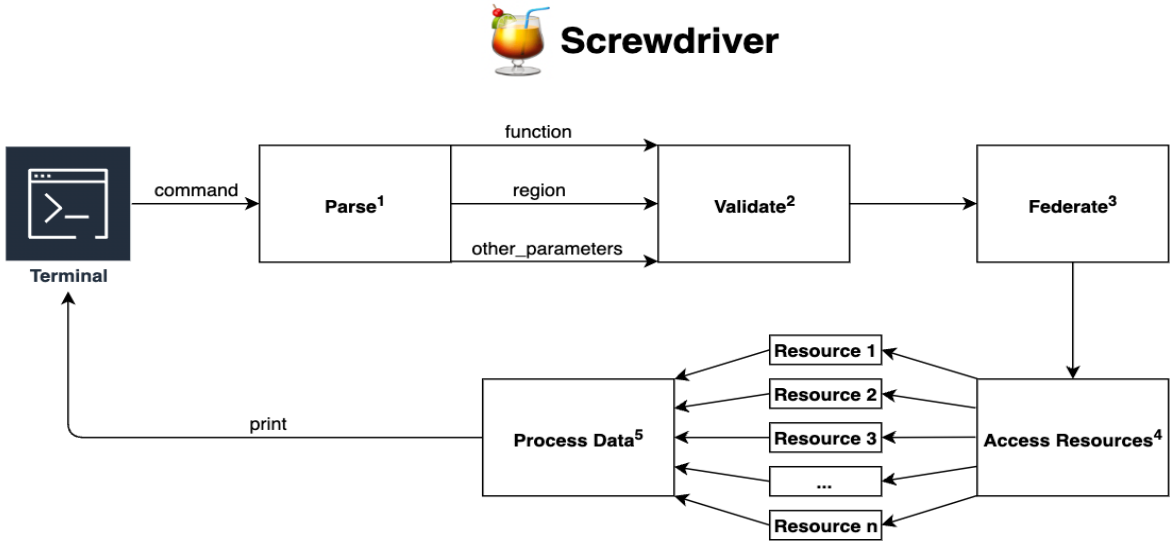

Figure 1: Design of Tool

3.1 Parser

The first component is the parser. Users enter one of twelve supported commands (the syntax for which is outlined in documentation that I wrote) through the terminal, and the parser reads in their input and splits it up into different parts. Specifically, each command is split up into the main function that is being called (which is used to specify what data the user wants retrieved), the region (AWS services have various regions, such as *us-east-1*, that have to be specified so that the software knows where to fetch the data from), and any other parameters that the command requires (such as login credentials). The parser performs this process and stores the different parts of the command for later access.

3.2 Validator

The next component is the validator. Once parsed, each command has to be validated to ensure that the command was entered with appropriate syntax (i.e., format, including spelling). The validator checks to see that each part of the command is valid: the main function specified must be supported by the tool, the region specified has to exist, and each function should have any additional information it needs provided through the other parameters when the command is entered. If any part of the command is not valid, the validator prints out an error message and the tool requires the command to be re-entered.

3.3 Federator

The third component is the federator. This part of the tool automatically retrieves credentials stored in the user's computer, and uses them to log in to one of many administrative accounts. Each of these accounts gives AWS software developers on the CW RUM team access to customer data, which is what is being retrieved by the tool. If federation fails, the tool again prints out an error message and the user must re-enter the command.

3.4 Resource Accessor

The next component is the resource accessor. Once access to the relevant account is granted, the data requested (known based on the command entered) can be retrieved from the necessary resources. This includes accessing databases, running custom queries to search for data, and fetching basic information about a customer's account. The data retrieved is aggregated and returned for processing.

3.5 Processor

The final component is the processor. The processor takes all of the data retrieved by the resource accessor, and cleans it up as needed. This includes making the formatting neater, removing any unnecessary information, and rounding numbers if desired. With processing complete, the last step is to simply print out the information to the same terminal that the user used to enter the command in the very first step. The tool has now completed one full iteration of a command run and is ready to repeat this process for any future commands entered.

4 Results

The software that I developed is currently being used by a team of roughly 15 CW RUM software developers to resolve customer issues more rapidly. The tool allows AWS software developers to access information about customer accounts and about the service we provide programmatically and securely. The average time taken to solve a customer's problem has not been formally measured, but by account of all users of the tool, it has fallen significantly, in turn increasing customer satisfaction. Lastly, there are future plans in

place to make the software available to multiple other AWS teams for their benefit.

5 Conclusion

The tool I developed during my time at AWS serves as a mechanism for quick, reliable, and secure access to critical information about the CW RUM service and about customer accounts. It provides significant time savings towards issue mitigation and eases the onboarding experience for new hires on the CW RUM team, which eventually translates to a much better customer experience. The tool has been deployed to other teams so that its benefits can be fully taken advantage of. I am fortunate to have been a part of such an innovative team and am glad that I left my mark as a meaningful contributor.

6 Future Work

There are additional features that the team desires to add to the tool to extend its functionality. When returning to the team full-time in August 2023, this responsibility will likely be placed on me. The new features seek to make the tool even more insightful with regard to the data that it provides, and perhaps quicker as well. Once improved, the tool will likely be deployed to many other teams at AWS so that it can be used by dozens of engineers on a weekly basis. Beyond this point, the tool will need to be maintained, which primarily consists of adjusting functionality based on user feedback and fixing any bugs that are found.

References

[1] Beale, C. 2022. The importance of customer service. Retrieved September 7, 2022 from https://researchworld.com/articles/the-importance-of-customer-service

[2] Sheth, J. and Mittal, B. 1996. A Framework for Managing Customer Expectations. Retrieved September 7, 2022 from https://www.researchgate.net/publication/226707807_A_framework_for_managing_customer_expectation