

Designing an Automated Ticketing Service for Improved Efficiency and Usability

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

Vi Nguyen

Fall, 2021

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

Daniel Graham, Department of Computer Science

Designing an Automated Ticketing Service for Improved Efficiency and Usability

CS4991 Capstone Report, 2021

Vi Nguyen
Department of Computer Science
University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
vpn4ydf@virginia.edu

ABSTRACT

As a software engineering intern for Bloomberg LP under the Internal Applications Department, I modernized an automated ticketing service for improved efficiency, maintainability, and usability. My work involved migrating the service from Sun and IBM servers to newer Linux servers. This change provided several benefits, including cheaper server costs, increased speed, and reduced power consumption. I also simplified the software's logic that translates incoming information on clients and their login override requests into human-readable support tickets. Throughout the process, I worked closely with the Bloomberg Customer Support (BCS) team to determine the expectations for the service's functionality. These changes better support BCS representatives that respond to tickets as well as improve the maintainability of the service itself for future developers.

1 INTRODUCTION

This past summer, I interned at Bloomberg LP under the Internal Applications Department. I worked on redesigning and building OverTicketSvc, an application service that supports login to the Bloomberg Terminal.

Recently, there has been a company-wide push to move older services, like Overticketsvc, to newer servers. When modifying services, software engineers will typically make other updates to improve the services' performance. Overticketsvc specifically had outdated functionality, which was no longer used in practice. By reducing unnecessary logic, the system's design was streamlined and its overall reliability was improved. I worked directly with BCS team members to identify deficiencies and correct them in the system.

2 RELATED WORKS

This project centers on system design, specifically working with end-users to

improve the system's functionality. Previous research has explored the benefits of collaboration with users in design. Buurman (2010) concluded that "iterative, user-centred design . . . instead of the common, technology and market driven one, leads to more useful and usable products." In particular, adapting software as user needs change over time is a critical part of the development process. According to Pahl (2004) "continuous involvement of users is as important as the constant evaluation of the system and the observation of evolving environments."

3 PROJECT DESIGN

The project began with a research and documentation phase. During this process, I familiarized myself with the existing system. The terminal's login process requires biometric authentication. However, for technical issues, users may request a login override that replaces the biometric check with two-factor authentication. When a login override request occurs, OverTicketSvc sends a support ticket to the Bloomberg Customer Support (BCS) team. Team members can later follow up with clients to resolve issues with the login process.

To move to application to a newer server, I determined which of the service's functions were incompatible and needed to be replaced. I also identified areas of logic that could be simplified. To best serve the sales representatives that use Overticketsvc, I worked closely with the BCS team

throughout the process of designing and testing the service. The BCS team contributed to major design decisions concerning the workflow of handling a request. For example, when a login override is requested, Overticketsvc will either create a new support ticket or update an existing one. If another login override was requested within the past 30 days for a similar reason, the service will add additional information to an existing ticket. This functionality streamlined the process of reading tickets since all relevant information will be collected in the same place.

In terms of implementation, I wrote a new client to call ticket creation and modification APIs. This functionality replaced many of the outdated function calls that were not compatible with the Linux servers. I also reworked the internal function calls to better align with the workflows of the BCS team. Finally, I converted the build to CMake and updated the Continuous Integration/Continuous Deployment (CI/CD) pipeline. These changes streamlined the testing and deployment process, improving the service's maintainability.

Throughout the project, I performed incremental testing on the service to verify that it maintained its functionality. The initial testing that I performed was unit testing on individual functions. I later ran system tests that checked the service's ticket creation and modification given set inputs. Before the service was deployed to production, I also worked with a testing team to perform User

Acceptance Testing (UAT). This phase was particularly helpful to identify bugs that a client, as opposed to an internal user, might encounter.

4 RESULTS

The service is advantageous for both the users and sales representatives on the BCS team. For users, they no longer must call the customer service line for login issues, reducing the time that they would be locked out of their accounts. For the BCS team, sales representatives can easily access a record of known login issues associated with an account. The number of phone calls that they must answer is also reduced. This functionality was especially advantageous during the beginning of the COVID-19 pandemic when a significant number of clients began working from home and encountered login issues from their work-from-home setup. The service currently handles over 5,000 tickets daily and is used on a global scale.

Migrating the service to newer Linux servers provided several benefits, including cheaper server costs, increased speed, and reduced power consumption. In particular, the servers improve energy efficiency by nearly ten times. Modifications to the system's business logic also improved the software's overall maintainability and readability. Streamlining the ticket creation process provides key benefits to customer support representatives.

The updated service makes responding to received tickets faster and easier.

5 EVALUATION

Throughout the internship, I gained experience with several new technologies. Before joining my team, I underwent a 1.5-week training program to gain familiarity with common technologies and frameworks used. The training also covered Bloomberg's internal application system, including how to write, test, and deploy software for the Bloomberg Terminal. Through my project work, I applied the skills and knowledge that I had learned from this training period. An essential skill that I relied on extensively was the ability to effectively search through, understand, and use documentation.

I also had the opportunity to develop soft skills that are essential to working in a professional team environment. By participating in daily meetings, I gained practice with effective communication and asking questions. Through my meetings with BCS representatives, I gained experience working with individuals from nontechnical backgrounds and translating their feedback into actionable tasks. I also had the unique opportunity to work with teams based in other countries and gained insight into coordinating collaboration on a global scale.

Of the computer science coursework that I had taken, the most helpful classes were CS 2150 Program and

Data Representation and CS 3250 Software Testing. The programming assignments in CS 2150 provided experience with working on open-ended problems, clarifying ambiguities, and implementing solutions in code. This course also emphasized understanding the underlying functions of data structures which is essential when optimizing a system.

When writing code in industry, software testing is crucial to ensure that users do not encounter errors. The course CS 3250 provided valuable experience in writing effective test cases that sufficiently cover an input domain. This class also introduced the method of Test Driven Development (TDD), which I applied in my project.

I am also currently enrolled in CS 3240 Advanced Software Development Techniques this semester. This course discussed the Agile development methodology, including the process of writing user stories, assigning points to these stories, and working in sprints. During my internship, I worked as part of an Agile team and saw firsthand these techniques used in action. Although the methodology is relatively easy to learn, having exposure to this information beforehand would have been helpful. This course also allows students to gain valuable practice working on a team project and using version control software to collaborate on code.

The Computer Science curriculum might be improved by including system design. When working on a large-scale service, software engineers will typically make several decisions regarding the infrastructure of the service and determine which frameworks to use. In evaluating options, software engineers must consider the trade-offs and benefits of each. In an academic setting, students generally do not encounter such decisions and typically have the mindset of simply writing code that works rather than writing code optimally. A course that included aspects of system design could be beneficial to students planning to pursue a career in the software engineering field.

5 CONCLUSIONS

This project was completed to improve the functionality of Overticketsvc, an automated ticketing service to report login override requests. By working with the BCS team, I redesigned the service to best fit their needs. As a result of collaborating with the end-users, the improved service offered several measurable benefits in terms of speed and efficiency while also easing the workload of customer support representatives.

6 FUTURE WORK

This project focused on improving the backend functionality of a service. Future work could include updates to the application's user interface of the application, specifically exploring user interaction with the system.

7 REFERENCES

Buurman, R. D. (2010). User-centred design of smart products. *Ergonomics*, 40(10), 1159–1169.

<https://doi.org/10.1080/001401397187676>

Pahl, C. (2004). Adaptive development and maintenance of user-centric software systems. *Information and Software Technology*, 46(14), 973–986.

<https://doi.org/10.1016/j.infsof.2004.04.004>