

**WAYFIN CST: BUILDING A HELPFUL AND USABLE
CUSTOMER SERVICE DASHBOARD**

A Research Paper submitted to the Department of Computer Science
In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science in Computer Science

By

Austin Houck

April 27, 2022

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.

ADVISOR

Daniel G. Graham, Department of Computer Science

WayFin CST: Building a Helpful and Usable Customer Service Dashboard

CS 4991 Capstone Report, 2022

Austin Houck
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
akh4rf@virginia.edu

Abstract

As a member of the Wayfair Financing team, I had a backlog of 157 outstanding ops tickets to deal with. As many of these tickets were simply the result of customer service representatives (CSRs) not having access to proper information in a concise, usable format, I was tasked with building a tool that allowed CSRs to quickly resolve these issues before reaching the engineering team.

I utilized my skills in software development, employing technologies such as PHP and MySQL on the backend and React.js on the frontend, tools which make up Wayfair's software stack. I created a customer service dashboard tool with the intention of making the most important information prominent, enabling CSRs to more effectively and efficiently assist customers. Initial feedback from customer service management was positive, and led to the creation of the "Customer View" button, which enables CSRs to see exactly what the customer saw at the moment issues started occurring. Although the tool needs a few more adjustments, as well as additional recommendations from the customer service team, this tool has the potential to dramatically improve the effectiveness of CSRs in helping customers, leading to a better overall customer experience, reduced frustration from CSRs, and more time for engineers to work on other tasks.

1 Introduction

To employ an effective customer service operation at any company or organization, service representatives need to be provided with information relevant to any

issue they may encounter. Further, this information must be presented in a simple, user-friendly manner that avoids creating confusion for both the representative and the customer. Unfortunately, the extent to which this need is met varies widely from company to company, with some customers having to wait on hold for long periods, only to be told that their representative doesn't have enough information to help them. Within the realm of technical issues and bugs with a company's apps or services, these customer complaints frequently end up on the desks of the engineers who built the products, burdening them with hundreds of menial troubleshooting tasks and rendering CSRs essentially pointless in this process.

At Wayfair, I was tasked with finding a solution to this problem, which had quickly grown out of hand due to many users having problems with our Wayfair Financing tool. When I arrived, there were over 150 ops tickets related to this issue, many from customers who complained that they weren't being offered any quotes, or that quotes they had been offered previously were no longer showing up. The engineering team had access to all logs and user information needed to troubleshoot and discern why these issues were happening. We had been contacting our vendors to resolve issues, but it was extremely inefficient to have the engineering team do tasks normally expected of customer service. Because this issue had been such a bottleneck for the engineering team, I was tasked with creating a tool that allowed CSRs to troubleshoot using the same data the engineers used, albeit in a much more understandable

format. Through consultation with CSRs, customer service managers, and engineers on various teams, I developed the Wayfair Financing Customer Service Tool, an interactive dashboard that displayed all relevant information every time a customer requested a quote to finance their purchase.

2 Related Works

Knowing core usability concepts regarding the design of web applications is critical for the development of a truly user-friendly experience. One source that I relied on is from one of my professors, Upsorn Praphamontripong [2021], who gave a very informative presentation on the subject. This presentation has frequently informed my perspective on designing applications for usability, including the points about making the most important information stand out, reducing wait time as much as possible, and requiring as little foreknowledge of the system as possible, all of which were relevant to the design of the Wayfair Financing Customer Service Tool. I also frequently refer to the WCAG Accessibility Checklist to ensure that I am designing frontend applications that anyone can comfortably use [2].

3 Project Design

In approaching this problem, there were several challenges that needed to be addressed. These included figuring out which data in our database was necessary to display, the format in which to display them, and resolving conflicts between components from two separate repositories. Additionally, we decided early on that we wanted to allow customer service representatives to be able to see exactly what the customer saw when problems began occurring, as we believed this would assist the representatives in understanding the nature of the issue.

Customer Details [Get Link](#)

First Name: [REDACTED] Last Name: [REDACTED]
 Customer ID: [REDACTED] Email ID: [REDACTED]
 Date First Applied: 2020-10-15 00:22:11 Date Last Applied: 2021-05-24 13:26:01
 Active: ✔

Customer Application History

View By: Lender Name Request ID Offer Type Show Filters

5497AC2-3AB2-38E6-38AE-F4F8B8CBAD4D

Order Id	Customer Type	PREQUAL_CUSTOMER													
Lender Name	Vendor Name	Offer Status	Present Order	Created on	Offer Expiration Date	Offer Type	Approved Amount	Display Offer	Lender Offer Token	W/Cc Pathdown	Is First Lock				
Fortivo Retail ...	ChargeAfter	NEW_O...	1	2021-05-24 13:26:01		LINE_OF_CRE...	\$2000	✔	gAAAAABye...	✘	✔				
Fortivo Retail ...	ChargeAfter	NEW_O...	2	2021-05-24 13:26:01		INSTALLMENT...	\$2000	✔	gAAAAABye...	✘	✔				
Genesis Credit	Genesis Credit	NEW_O...	3	2021-05-24 13:26:01	2021-06-30 13:26:01	LINE_OF_CRE...	\$5000	✔	6d3H5av0-6...	✘	✘				
Genesis Credit	Genesis Credit	DECLIN...	1	2021-05-24 13:26:01	2021-06-30 13:26:01	DECLINED		✘		✘	✘				

Page 1 of 1 20 rows

Figure 1: Original Tool

3.1 Removing Unnecessary Data

I began this process with a tool that simply output all of the data we had for a given application, shown above in Figure 1. Clearly, there were several issues with this design, chief among them being that there was so much data being presented to the user that the data in some cells was being cut off. To address this, I communicated with my coworkers and a customer service manager to determine which data was unnecessary and which data still needed to be added. In doing so, it was determined that not only were several columns either redundant or simply not relevant to the needs of customer service agents, but some of it was actually wrong or inconsistent, with the “Present Order” column in Figure 1 being a prime example (two different offers cannot both be first in the list). Additionally, the “Lender Offer Token” column was essentially gibberish to anyone that was not an engineer. Removing these faulty columns gave the data a lot more room to breathe and made the tool easier to understand.

3.2 Improving Data Presentation

Next, I focused on further streamlining the user interface (UI) of the tool in order to further reduce clutter and allow service representatives to find what they were looking for more efficiently. First, I realized that it would be far more helpful to identify each application by the date and time when it was submitted, rather than the application ID, which was an incoherent string of characters. Thus, I removed the “Created on” column from the table, instead

placing the timestamp in the leftmost section of the application dropdown header and moving the application ID to the rightmost section of the header. The font weight of these details were also increased to make them more prominent. These changes were made to bring the date and time to the representative's attention first, as English-speakers interpret information from left to right.

The next improvement I proposed was adding a button that, when clicked, showed the service representative the same financing offers that the customer was presented with at the time when issues began occurring. Doing so allowed me to move information such as offer amount, APR, and whether the offer was shown to the customer to a popup that displayed the offer cards so that representatives would only see this information if they needed it, further reducing clutter. I made sure to adequately draw attention to this "Customer View" button by making it bright purple and putting it in the top left corner of the dropdown, one of the first places a CSR's eyes might land. These changes are shown in Figure 2 below.

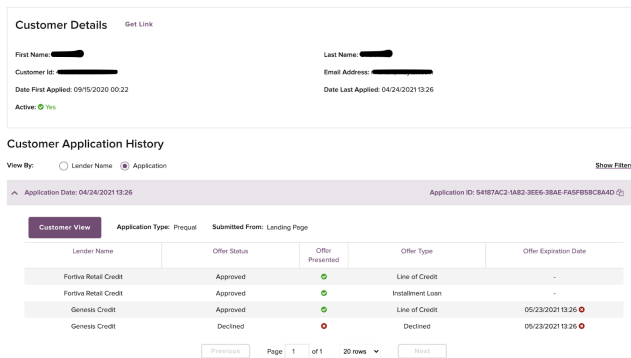


Figure 2: Customer Service Tool After Improvements

3.3 Issues With UI Components

To implement the "Customer View" popup, I intended to use the same UI component that was used in the customer-facing application, simply feeding the component the necessary information while retaining the exact same structure and style that the customer saw when they were applying for Wayfair Financing. Unfortunately, I found that apparently, only customer-facing pages had access to the Offer Presentation components, which meant that my

admin-facing tool was unable to directly use these styles. This was a nuisance for a couple of days, though eventually I was able to find a workaround. This involved creating a separate script in the customer-facing repository, and embedding the associated page within an HTML iframe component in the "Customer View" popup, thereby allowing the tool to indirectly use the component. Coordinating this was a bit of a hassle, as it involved changes to multiple repositories, which are not necessarily merged into production at the same time, and thus for a short period of time the tool was inoperable. Thankfully, my changes were successfully merged, and the tool worked as intended.

4 Results

Overall, this project was a success, as it provided a solid foundation for developing better tools oriented towards the needs of customer service representatives at Wayfair. While my time at Wayfair ended before I was able to see the full effects of the introduction of this tool, initial reports from the customer service department were favorable, with one customer service manager and multiple CSRs praising its simplicity and usability. Since my departure, one of my former coworkers has informed me that the number of outstanding ops tickets regarding issues with Wayfair Financing offers went down after the tool's introduction.

I suggested future improvements to be made to the tool should I be reassigned to the Financing team upon my return to Wayfair full-time in the summer of 2022. One such improvement that I would like to implement is a feature that provides CSRs with mechanisms to contact various lenders directly to resolve issues pertaining to application login in external systems, rather than having them fill out an ops ticket and ask the engineers to do so. If given the chance, I would likely approach this by adding some sort of button or icon next to the lender name that, when clicked, would automatically generate an email or Slack message to the lender depending on the type of issue being experienced, populating the message with the relevant information such as customer ID, application ID, and the timestamp of the application. This would further reduce the number of ops tickets

that engineers have to deal with and would save time and frustration for all parties involved. Additionally, I would like to look into ways to improve the performance of the “Customer View” popup, as sometimes it takes a few seconds to load the relevant offers, which can be frustrating to deal with.

5. Conclusion

In summary, the new dashboard tool outlined in this report will improve the effectiveness and efficiency of both engineers and customer service representatives alike. By enabling representatives to see what a customer saw when their problems began occurring, and providing a user-friendly interface to access the customer’s relevant information, Wayfair’s overall customer experience will be greatly improved. More feedback will be needed in order to ensure the best possible cohesion between these representatives and the tool, but the work done in this project serves as a step in the right direction.

6. Future Work

As mentioned previously, there is definitely more work that could be done to improve the usability and functionality of this tool. Along with a mechanism for service representatives to directly contact lenders and some tweaks to the “Customer View” popup, there are other possible areas of improvement. One such example is ensuring that the way in which the workaround for the “Customer View” popup was implemented doesn’t create any security issues within the site. Additionally, some simple query optimizations for the tool would go a long way in improving the performance of the tool and reducing excess strain on our databases.

7. UVA Computer Science Program Evaluation

Several classes at UVA prepared me for my work on this customer service tool. In particular, taking Advanced Software Development helped me improve my skills in project management, and serving as my team’s requirements manager in that class prepared me for the process of eliciting feedback from service representatives and other relevant stakeholders. The

class also introduced me to the MVC framework, which was directly relevant to my work at Wayfair.

References

- [1] Upsorn Praphamontripong. 2021. Users and Usability Principles. Retrieved February 20, 2022 from <http://www.cs.virginia.edu/~up3f/cs4640/slides/4640meet03A-UsabilityPrinciples.pdf>
- [2] World Wide Web Consortium. 2019. How to Meet WCAG (Quick Reference). Retrieved February 20, 2022 from <https://www.w3.org/WAI/WCAG21/quickref/>