

CASE STUDY

Expanding features for CRM systems products, part 2

INTRO

AUTOMATIC ACCOUNT SEARCH

KNOWLEDGE BASE

APPLICATION FULLFILLMENT

OUTAGE PROCESS

Automatic Account Search

Why prioritize this feature?

My main priority was to make the customer service and sales agents more efficient. After conducting user research in which I timed their tasks with a stopwatch, results showed searching for a customer account took 30% of the average call time. This was the most time intensive task, outside of understanding the callers problem and finding a solution. If we could speed up this task, customer service agents could answer more calls, and we would theoretically reduce customer hold (15 minutes) and call times (8 minutes).

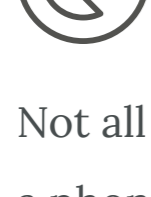
User story: "As a customer service agent I want to be able to find customer's accounts when they call without asking them for information, so that I can focus on empathetic listening and problem solving."

What is auto account search?

Auto account search was a new product feature to automatically find and show the customer's account in our CRM System when the customer service agent answered each call. Using the customer's inbound calling number as the string for the search query, made sense, as it was the only information accessible before a call was answered.

Challenges

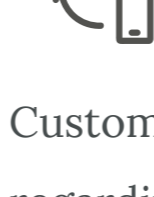
While this idea may sound simple in it's conception there were quite a few obstacles to overcome:



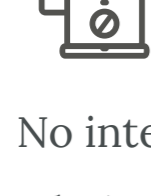
Not all customers would have a phone number on file



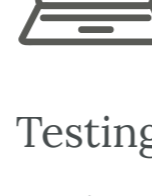
System was setup for only one phone number per customer account



Customers could be calling regarding an account with one or more of our 11 brands



No integration between the admin site (CRM) and phone system



Testing and deploying against 11 different admin sites



Edge cases like international phone number formats

Other approaches and considerations

Another consideration was not building this feature. The feature scope included integrating 2 systems and creating a custom search algorithm across all 11 different databases, and would have user experience implications. One option was only building this feature on our top 2 brands, but if we had to do the work for 2 brands, the work to scale it to 11 was marginal comparatively. In the end, my team and I agreed to build this feature for all 11 brands because the expected benefits in the end user experience and increased customer service efficiency outweighed the implementation challenges.

The solution

The automatic account search feature was built in house. The majority of the heavy lifting would be on the backend, so I created a mini PRD with user stories, interface mocks and technical specs to empower my engineers with enough information to get started.

The solution for this new product feature was pretty straightforward from a UI/UX perspective. The Admin sites (CRM) interface was accessed via the web browser. It was logical to open a new tab in the end user's default browser, with the results of the search query using the customer's phone number. This would happen every time a customer service agent answered a call. This approach helped overcome most of the challenges and difficulties listed above.

There were 3 primary user (customer service agent) flows for auto account search:

- The first user flow was the ideal scenario, where a customer calls from the phone number on their account, and the auto account search feature would retrieve and present the customer's account to the customer service agent. This required no design as the account page already existed.
- The second user flow would be triggered when the customer was calling from a phone number associated with multiple accounts or brands. I mocked a results page with the capacity to show a list of results based on the inbound calling number.
- The third and final user flow would be triggered if the phone number the customer was calling from was not associated with any account across any of the 11 brands. I repurposed the results page design to show a no results state. If this flow was triggered, the customer service agent would then manually search for the customer's account using a last name or email address, as done previously.

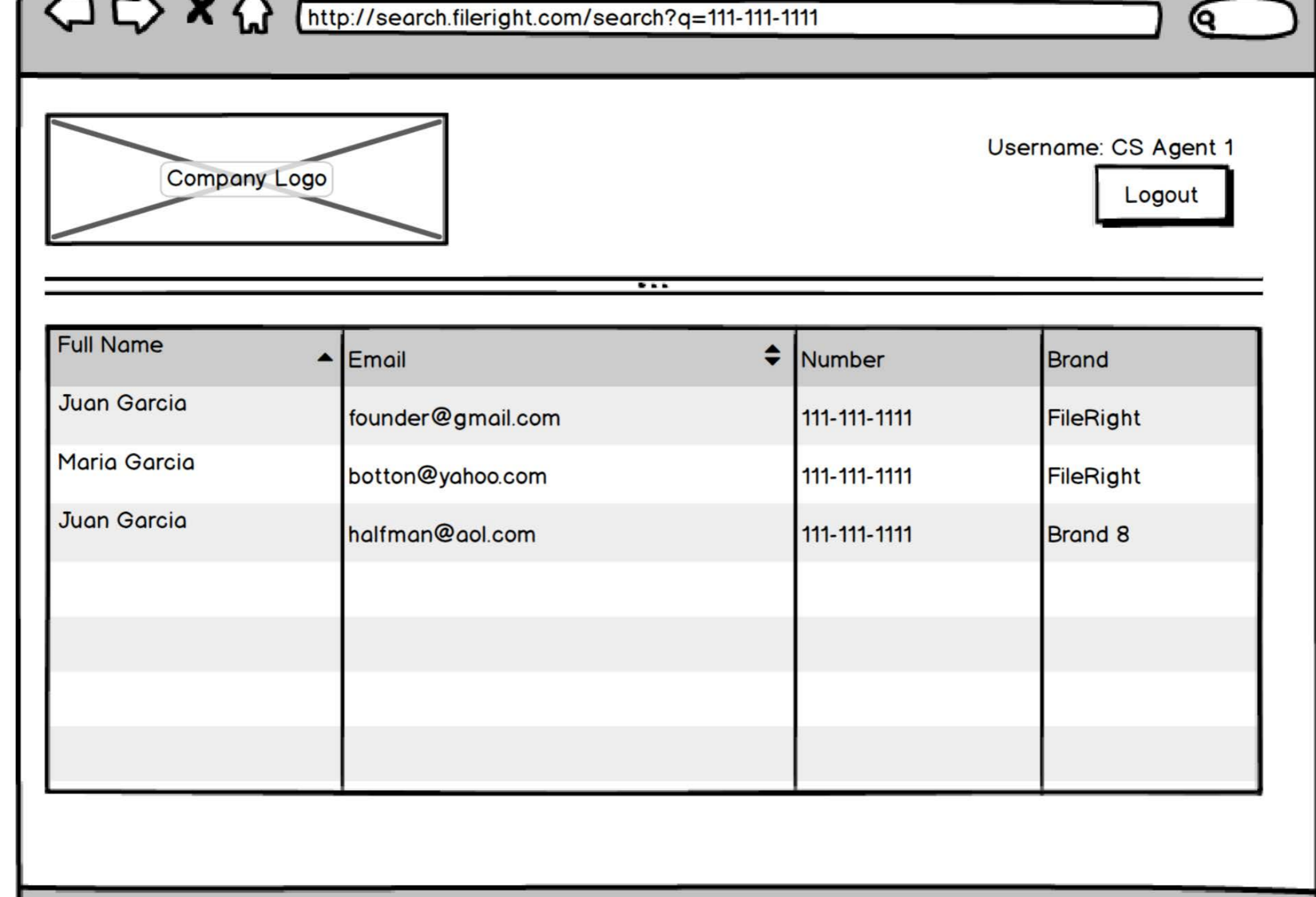


Figure 3. A wireframe I created for when the customer was calling from a phone number associated with multiple accounts

Development, QA, and launch

To ensure no major delays in the development of auto account search, I served as scrum master during daily stand up, and worked closely with engineering to remove any roadblocks. I prioritized shielding the engineers from new requests, and testing the first build on only 1 brand to keep the timeline on track. In the QA phase of the project, I thoroughly tested the feature on the floor while answering real customer phone calls, and prioritized all the technical and user experience bugs for resolution. After auto account search passed QA, I advertised and trained customer service agents on how to use the new feature. During launch week, I found a desk in the customer service department to work from, so I could answer questions and receive real time, unfiltered feedback.

Measuring success

The goal of auto account search was to make the customer service agents more efficient by automating one of their most time intensive tasks. I measured the results of the feature using objective, quantifiable metrics to see how they compared to the success criteria.

- 22x improvement in account search efficiency by reducing time to find a customer's account from 46 seconds to 2 seconds
- 72% of inbound calls automatically presented customer information to the customer service agent

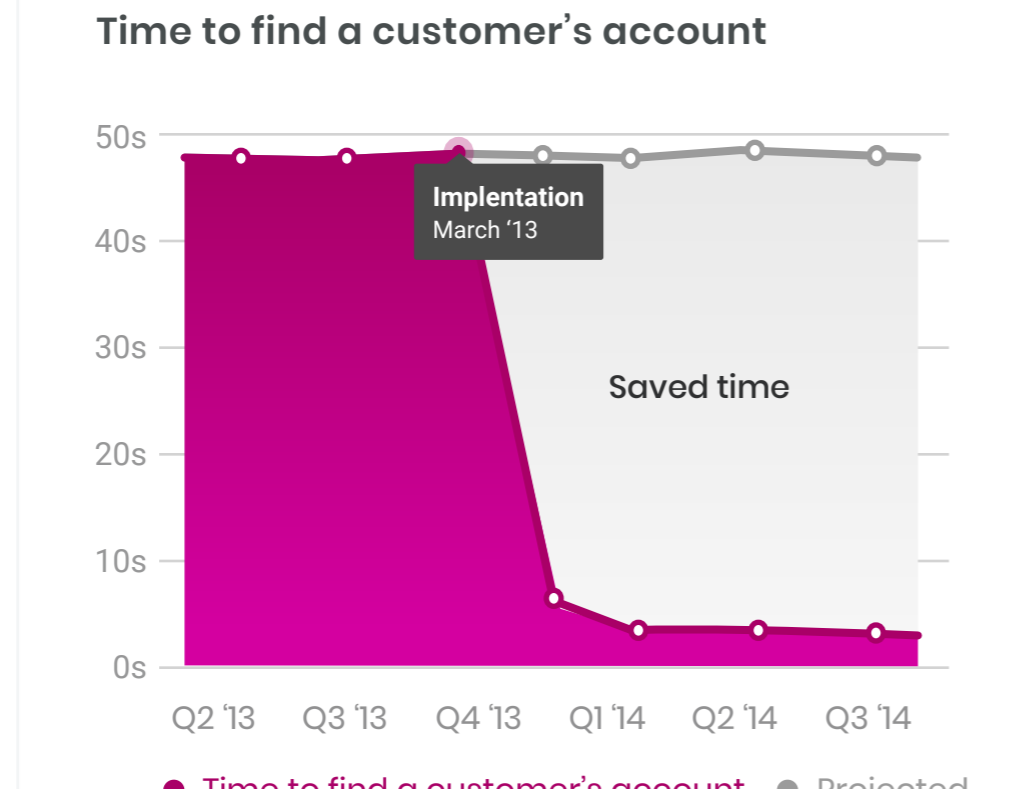


Figure 4. Time to find a customer's account

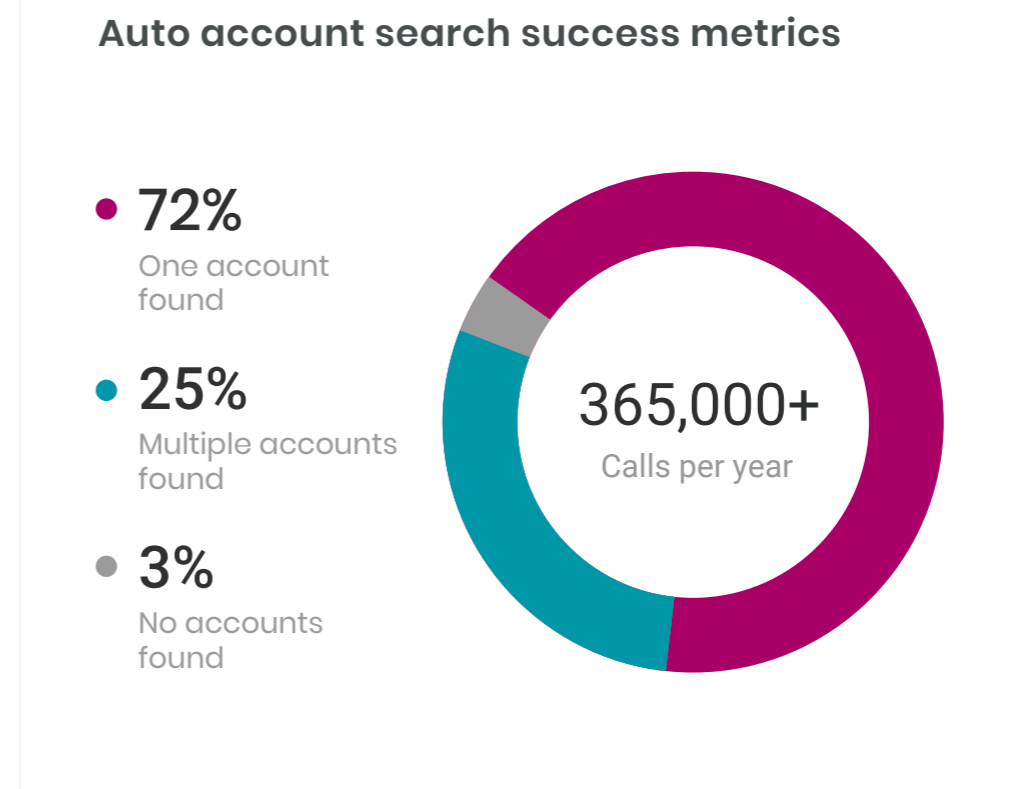


Figure 5. Calls per year & percent of accounts found

It was also important to collect some qualitative data points, as solving the problem with happy end users was key to building trust as a new product manager in the organization. Anonymous user surveys reported high approval ratings via a net promoter score and high ease of use ratings via a satisfaction survey.

“OMG this saves me so much time”

DIANA, CUSTOMER SERVICE AGENT

Phase 2 and next steps

Phase 2 of this feature would also extend auto account search to the sales department. However, the sales teams had other pain points that were prioritized ahead of this feature that would provide more value for their roles (more on this in the [application fulfillment](#) section).

Feature Brief for “Deploying Auto Search To Sales”

Background

Over the last 10 months, the customer service department has experienced increased call efficiency and productivity as a result of the deployment of auto account search. The auto account search feature alleviates the need for the customer service agent to manually search for customer accounts across the different admin sites. Currently, auto account search is only deployed to the customer service department.

Need

We want to deploy auto account search to Sales department, so they too can achieve increased call efficiency and productivity.

Problem with our existing product/service

The existing auto account search handles partial search and this works well for the customer service department, but the sales department has different needs than customer service.

- The sales department would like to remove the partial search feature for all of their

Figure 6. Example of feature brief for deploying Auto Account Search to the sales department

Solving a session timeout issue

A quick win with a big impact

During the several hours per week I initially spent engaging with real customers via the CRM systems, I discovered a session timeout issue that was negatively impacting customer service agents. Customer service agents' sessions were timing out after only a few minutes while listening to and understanding a customer's reason for calling. After agents were logged out, there was an additional 5 minute waiting period to log back in as the server was burdened by too many simultaneous logins. Upon discovery, I immediately made a recommendation to change the session timeout from 5 minutes to 8 hours (a typical customer service agent shift) and increase the amount of server resources (CPU), so no login even during peak periods took more than 2 seconds. This alone improved call center hold times during peak hours from over 15 minutes to under 4 minutes. Solving this session timeout issue would save FileRight over \$600k per year in perpetuity.

Conclusion

Auto account search was a user experience and financial success for FileRight. Between auto account search and solving the session timeout issue, the call center hold times improved 6.5x from over 15 minutes (during peak periods) to less than 1 minute. The resulting return on investment for the organization was a reduction of at least \$1,330,000 per year in call center cost of labor.

By freeing up call center agents, FileRight was exponentially closer to achieving their goal of supporting 4 million new customers. It also helped solidify the need for my new role in the organization and garnered some political capital to help with future updates. I was ready to tackle the next feature, onboarding new agents more quickly so the business could profitably scale to accommodate a 200% increase in business.

CONTINUE TO PART 3: KNOWLEDGE BASE

