AppFolio Support Experience

A case study on the design, research, and strategy work done to improve the support experience of AppFolio customers.

Sheriff Jolaoso

Project Brief

Our company was in a growth period, and people in various departments were now tasked to consider how to scale efficiently as we planned to grow our customer base.

For our customer-facing roles, this meant some creativity was needed to consider how to help customers solve their issues effectively without high-touch interactions when they may not be required.

All-the-while, we also had customer feedback that was indicative of a support experience that could use improvement for a variety of reasons.

This project was done to both improve the current customer experience based on feedback and address scaling our support team effectively.

For this project, I was the service & product design team of one with ad-hoc support. I used the Design Thinking methodology throughout this project, and this case study will recap key findings and synthesis work done to draw conclusions on how to address the customer and business needs.

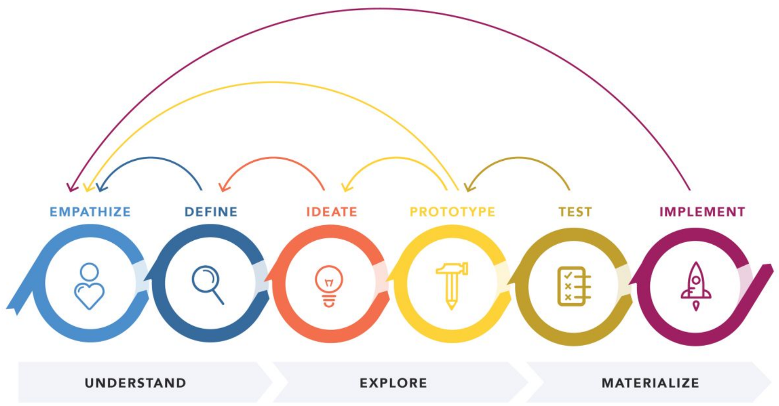


Diagram credit to NN/g

Empathize

At the outset of the project, I wanted to get a better sense of both the business and customer needs for improving the support experience.

There were two streams of empathizing to be considered:

From the business side:

- Current-state assessment of our processes and systems
- Evaluation of what solutions were available to us

And from the customer side:

- Customer interviews to better understand our customer base current sentiment of our support experience
- Non-customer interviews to understand the landscape of other B2B support experiences
- Competitive analysis to supplement what we learned from our non-customer interviews

Customer research findings

Through structured interviews with a set of our customers and a set of folks who used B2B software I facilitated, the team got a better understanding of the support experience from their point of view. Key aspects we learned were:

Internal support is often the 1st step

Interviewees in both our customer base and non-customer base mentioned that there were colleagues, sometimes designated as “super users” that they would go to first (or only) to get support.

This is a part of the support experience you can’t have metrics for, but understanding that it happens can give way to more opportunities for impact.

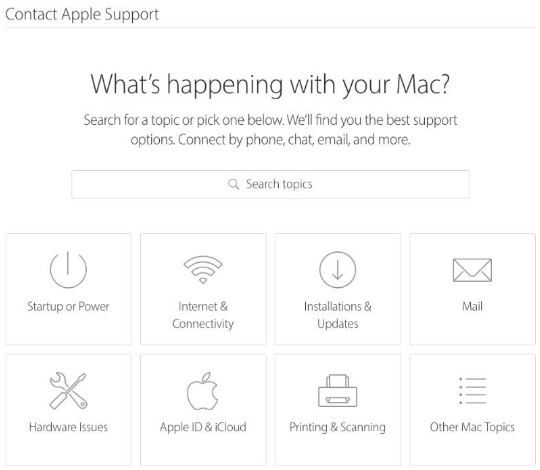
Discretion

An internal assumption was that if the phone number is too readily available, then users would call too often.

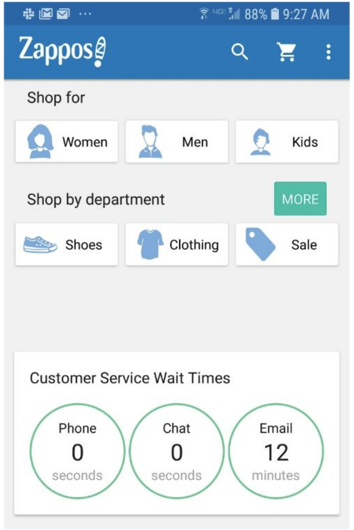
What we learned through interviews was that users are willing to a lot before reaching out or only reach out in the most dire situations and sometimes it doesn’t feel like that is acknowledged.

What could improve the customer support experience?

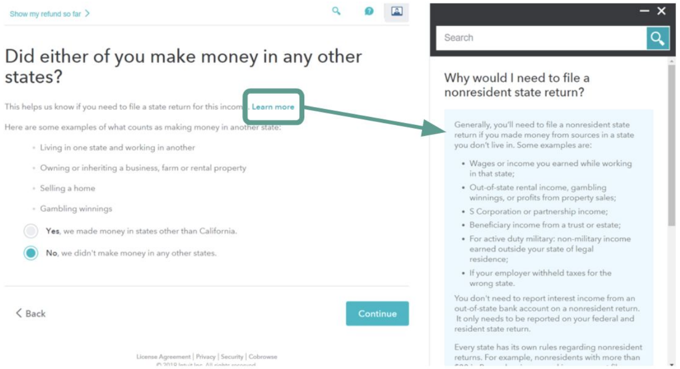
I did secondary research to understand what technology and orchestration of touchpoints worked well (and not well) for other companies who provided a support experience in the B2B & B2C space. Some valuable concepts we found for our consideration included incorporating:



Case diagnosis for routing



Wait time transparency



Help article search within the app itself

Internal interview findings

Through internal interviews within our customer success department & training department, I learned more about the current state of how we handled cases and what shortcomings existed within the process of handling cases and after the process:

Customers weren’t interacting with help center

For a mix of reasons, including the content density of articles and the fact that support requests would be replied to with help articles, a large subset of our customer base was not interacting with our help center prior to submitting a case.

Internally, it was anecdotally clear that our help content did have some value that could address a lot of cases that users submit without referring to our help center.

Case tracking processes were a bit opaque

Our support team has always look for opportunities to improve the support experience, but the improvements haven’t been orchestrated in a way to maintain a clean CRM.

This brings about issues when changes are made not considering the previous changes that have happened.

This only proliferated as more teams disparately make improvements to their teams’ processes alone without collaborating cross-functionally.

Define

With research in hand, I could start synthesizing and scoping what could be done based on the issues I learned about and the stories behind the issues.

I had a better understanding of:

- What did our users expect vs. what was popular on the market
- What was the root cause of case submission for our customers
- What could be done with and without technology to improve the experience
- How our customer needs could align with business needs

And more, which aided in the communication with directors in various departments about what the focus should be and why knowing there were a variety of things we could try to do.

From ocean to lake…

The challenge with improving the support experience is that it is multifaceted and there’s not just one central thing to change.

Just with the types of issues a user could run into in software, there’s a wide range of variability in the issues someone could run into. This makes the project so vast it can feel overwhelming.

But after talking to users and reviewing what they say, it became clearer there were some patterns in the story that we could look to address:

Troubleshooting > How-to

People expressed that normal help articles describe how to perform a task step-by-step, whereas when support is needed, it’s usually a troubleshooting need within the step-by-step process.

Learning from peers

A user’s first resort is not always us: they’ll go to a co-worker and user forums along with the effort of looking through help articles and other content before reaching out.

“You all have chat?”

Availability of communication options isn’t always readily known, which is a key user flow issue we realized.
A few of our own customers we interviewed were surprised we had a chat option as I asked about their usage of our different channels.

Internal cleanup

The user experience was not the sole experience we had to be diligent of. Through our internal research, we realized there were challenges in analyzing case data:

Support teams disparately using case fields: We have support teams in different departments that use case fields different than others. This results in noise in reporting every now-and-then.

Case field options being indicative of what another case field is better suited for: By way of not considering scaling, some of our case fields were being used in ways that other case fields already existed to handle.

The inability to go within a subset of cases within a case reason to identify common issues: We have case reasons like “Receipts” or “Work Orders”, and that alone does help create subsets of cases to focus on fixing. The challenge then becomes identifying more granular issues, especially those that happen most-commonly, so that can be communicated back to the rest of the company. Without that ability, there’s a bit of a “no-mans land” created, where we have data that can lead us to conclusions, but there isn’t enough resourcing to forage through the data.

Design Principles

As we continued to identify what would be most-important to address, the team decided that having principles we align our efforts to would be valuable, especially as we had more team members join in on the project. These design principles would provide scope to our project:

Recognize: Recognize the fact that a user has a preference for how they would like to get help and the fact that they may have done work prior to reaching out to a human for support. Also recognize the previous relationship established with the user with regards to their service experience.

Low-effort: Don’t make it hard for users to get the support they need, may it be human or digital.

Informed: When human support is needed, be sure to make sure the user is well-informed of the status & progress, and set proper expectations for the support experience, even if they are not the most-favorable.

Well-recorded: Keeping an up-to-date, detailed record of a user’s support needs will allow us to accurately & consistently report on their experience in order to identify opportunities for improvement.

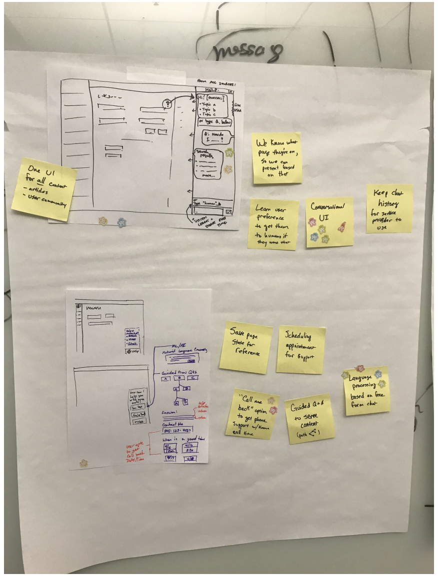
Ideate

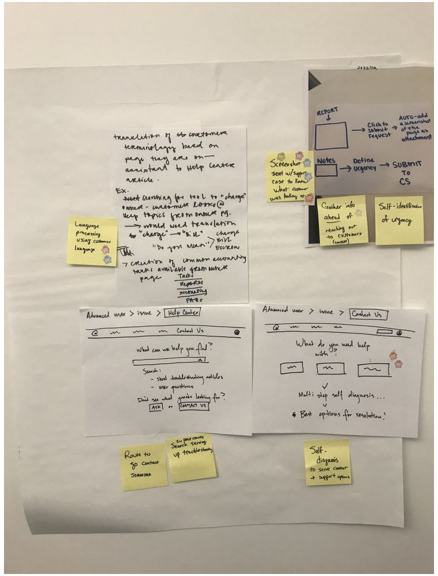
With defined scoped for the work, I could get jiggy knowing what we were looking to accomplish.

I was also in a better position to start collaborating with other stakeholders to get their focused input of what we could do.

Design studio

To start getting ideas from our internal experts, we did a design studio. We shared what was learned from our research and the design principles to provide guidance for our participants.





It was important for the creativity to come through along with the pragmatic viewpoints of what was truly possible from those closest to the experience.

Content & technical strategy

During the design studio synthesis and throughout the brainstorming process, practicality came into play when thinking about making an effective solution.

The digital solution to improve our user's support experience had to be applicable to the variety of areas of our app. If what we developed could only be useful for particular areas of or product, the investment may not be most beneficial to invest in.

Similarly, if the solution required our team to develop new content to help users through their issues, we wanted to be mindful of what that effort would look like.

So knowing all of that, we decided to focus on commonly-occurring issues our Customer Success & Training teams have identified and did a bit of a litmus test to see if we could make content. We also had technical feasibility conversations around the various ideas that came out of the design studio to evaluate what the effort would be to bring some of these ideas to life.

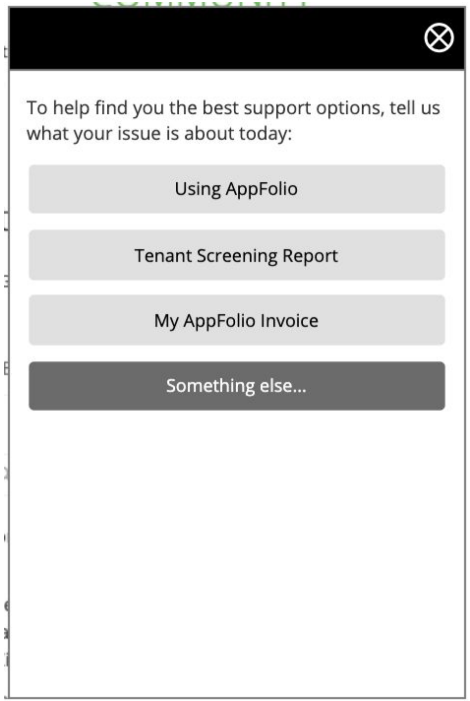
Prototype

Now with synthesized concepts, we had ideas we needed to validate with customers.

To do that, I developed interactive prototypes that would help the team get a sense of what direction we should go from a customer-facing standpoint.

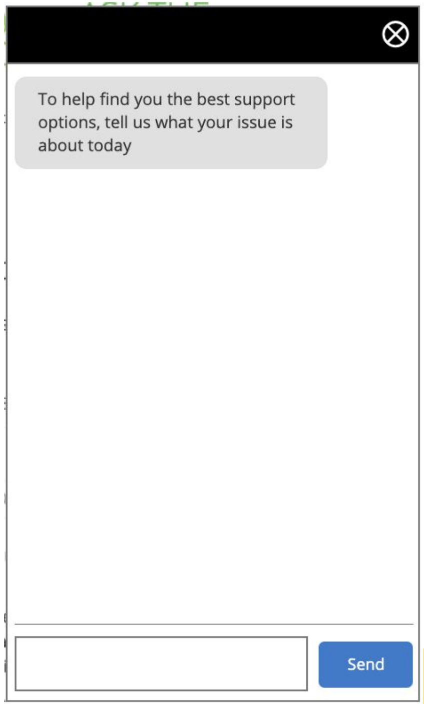
Prototypes

To evaluate ideas developed throughout the project, I prototyped 5 experiences for getting support based on all of what had been learned over the course of the project to this point:



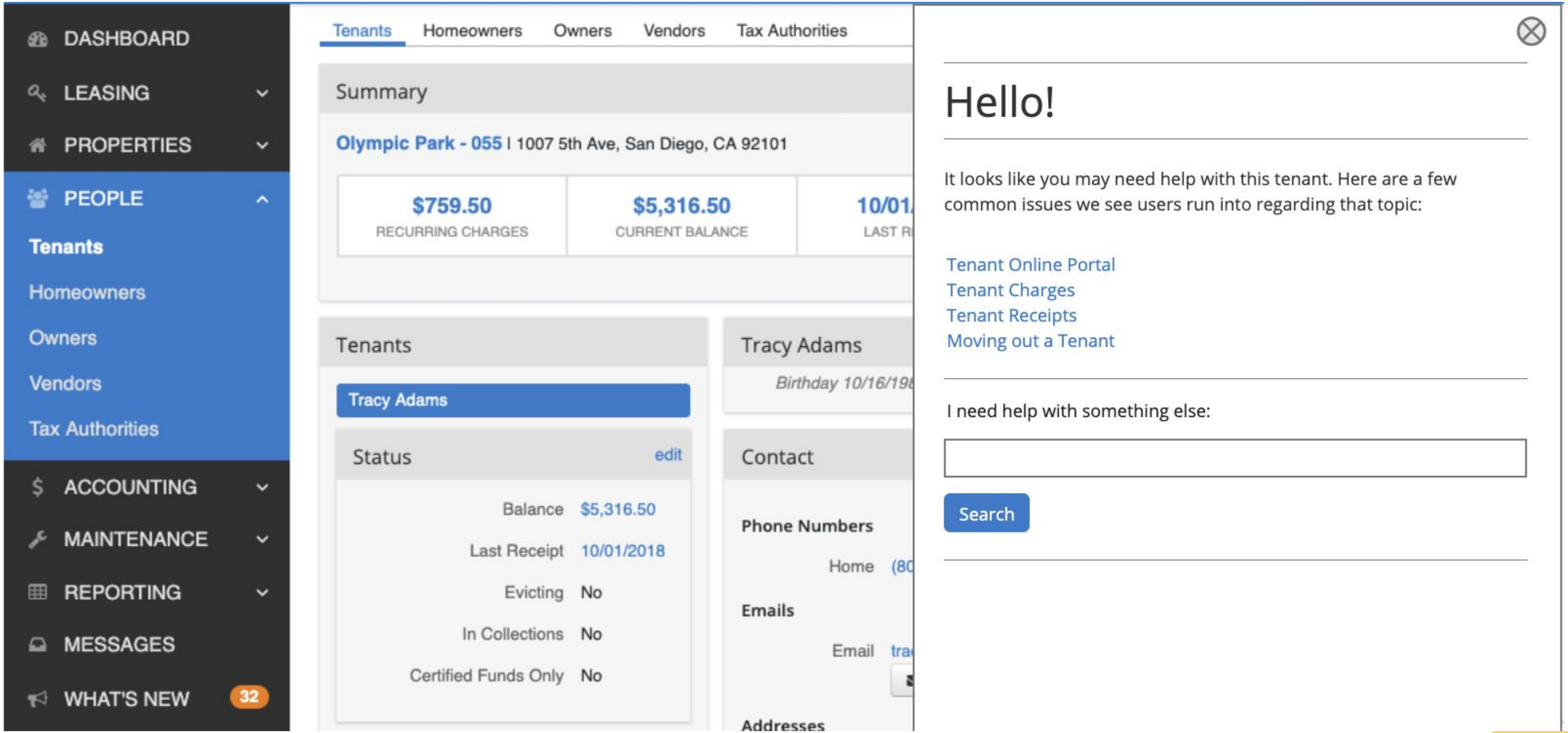
← Button-based chatbot
A guided chatbot experience where users clicked options as opposed to typing out their issue.

Conversational chatbot →
A chatbot experience where users typed out their issue.



In-app help panel

A context-aware help panel that would appear when “Get Help” is clicked within our application, as opposed to navigating the user to our help center.



Questionnaire

A case diagnosis questionnaire that would guide user to particular FAQs and additional support content

Using AppFolio > Tenant Management > Tenant Online Portal

Are you running into a problem with any of the following?:

Activating a tenant's online portal

My tenant cannot log into their portal

How can I see what my tenant sees in their portal?

How can I delete my tenant's scheduled payment?

My tenant made an accidental payment

My tenant made an accidental payment, or a payment in error

The tenant would need wait for the payment to process. Should the payment be successful, we recommend you refund them with a tenant payable.

Did this help resolve your issue?

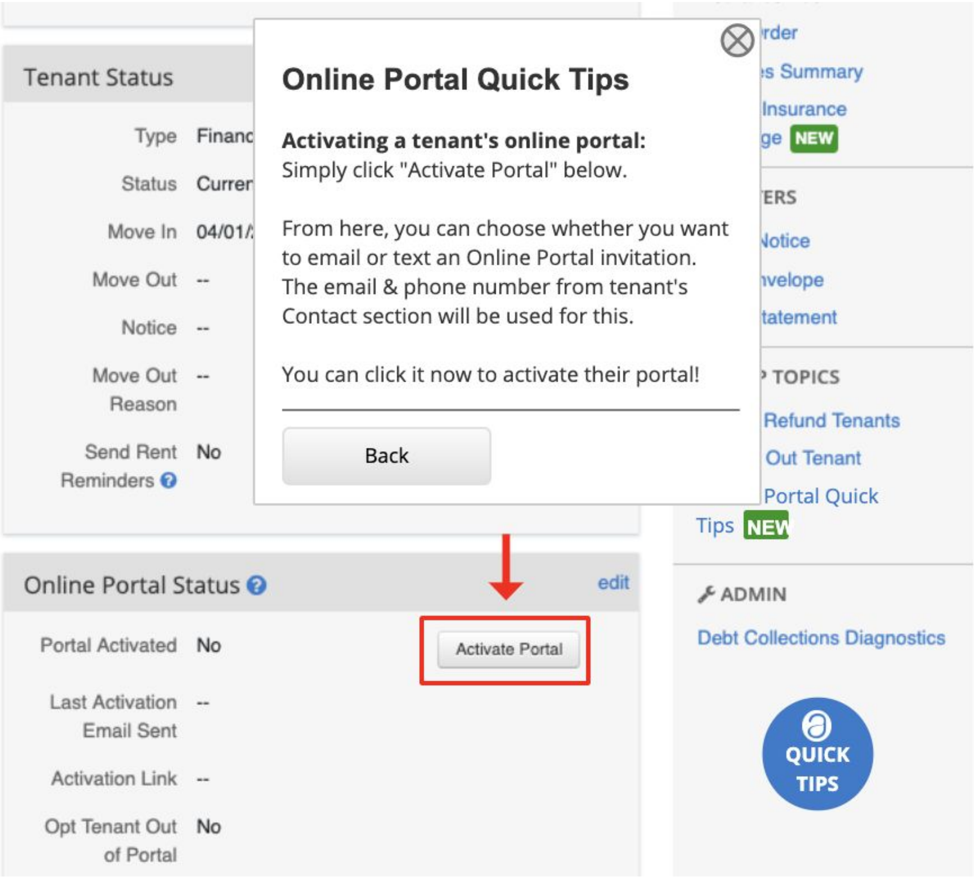
Yes

No

None of these are my issue

In-app walkthroughs

Feature walkthroughs that could guide users with structured guidance overlayed in our application.



Test

With prototypes in-hand, it was time to get customers' perspectives on what they felt would be best to use to improve their support experience.

Our test was primarily a preference test with our customer base. With the resourcing at-hand, we could not do full-fledged experimentation, so it was essential to make robust prototypes that felt "enough" like what the user's authentic experience was.

In the test, I showed the participant the prototype variants in randomized order and asked them their preference of which they felt would be most helpful if expanded to other use cases for issues they run into.

Usability test results

From usability testing, we found that the **in-app walkthroughs** were most preferred, but users and internal stakeholders alike noted that walkthroughs were not the most-scalable solution.

The option that we felt was most versatile, most favored by test participants, all-the-while could handle a majority of the issues we uncovered in our research was the **help panel option.**

Key results:

- Having help content open in the product window itself is already useful. Having to open the help center in a new tab can be relatively cumbersome.
- Users sometimes feel that we could describe their issue better than they could, which could be indicative of a few things we are reviewing.
- The chatbot experience was not one that more experienced users favored, and they shared concerns about it being in the way of getting in touch with a human, similar to a phone tree.
- A majority of participants understood the value of putting troubleshooting content in-the-way of submitting a support request.

Implement

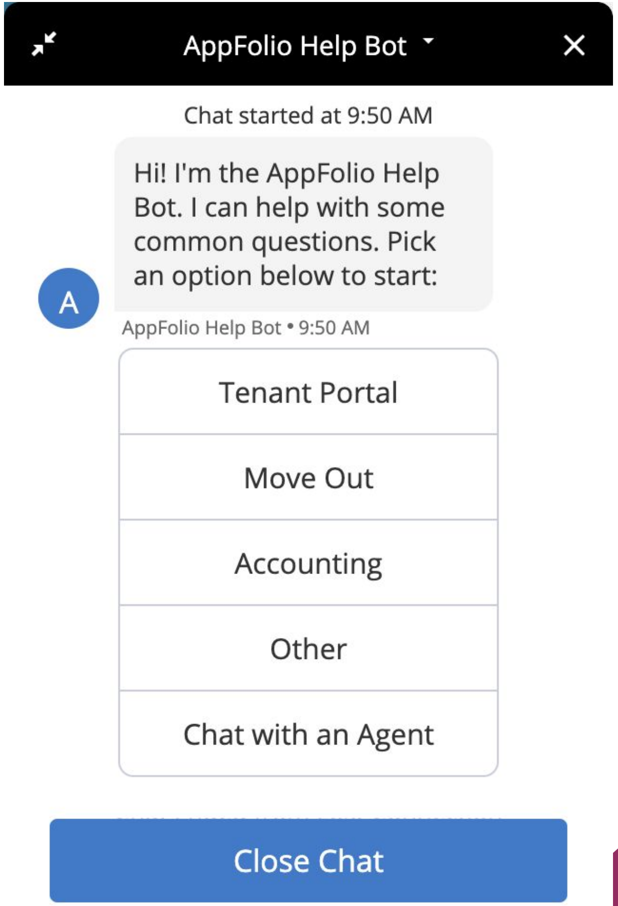
With understanding garnered throughout the entirety of the project, it was now time to decide how we would proceed with improving the support experience.

Chatbot...?

Although the chatbot was not validated as the optimal choice for customers, it was seen by stakeholders as the quickest way to deliver upon our customer and business needs as we did further technical analysis and identified resources to build the ideal solution.

We felt that with the chatbot, we could learn a lot about the efficacy of our snippet content as well as validate or invalidate hypotheses around what we felt were high-volume case driving issues that could be resolved by self-service means.

I developed the chatbot content and IA with our business systems team and the community manager.

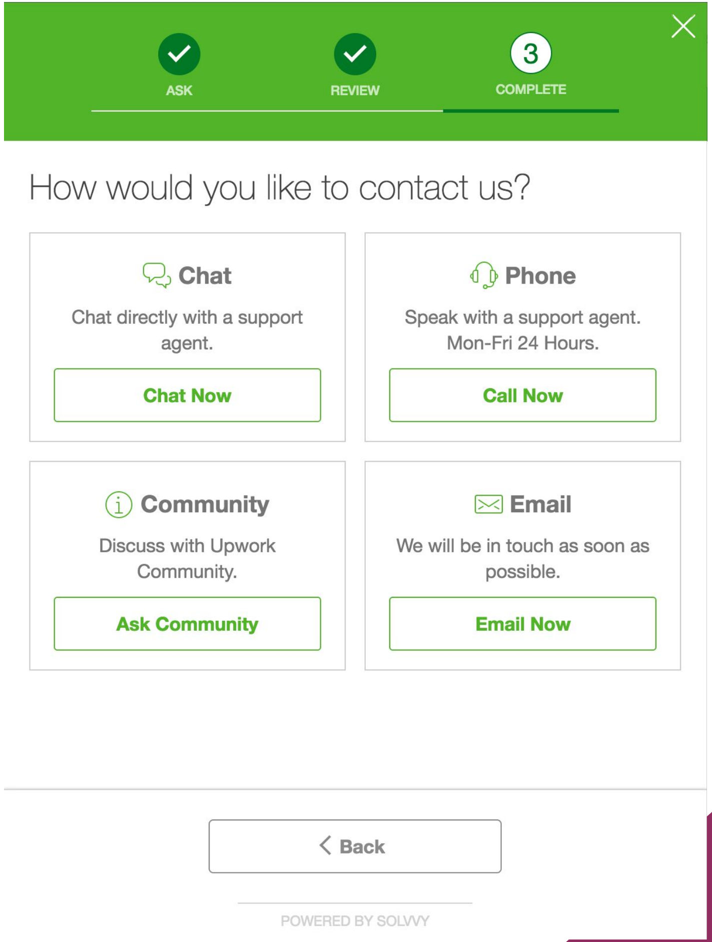


Help Panel

During the development of the chatbot, we did some additional vendor research and found an alternative that we felt could be just as fast to implement and provide the customer experience we validated.

Solvvy is a help panel tool that addresses a large swath of the issues we uncovered. It capitalizes on the currently-developed content to serve troubleshooting content and provides a clear view of contact options after a user has provided context around their issue.

Solvvy also mitigated the need for development resources we did not have at-hand yet.



What is going on today?...

I am currently in the process of implementing Solvvy for trial with our business systems team.

The team is also formalizing an explicit program will also encompass additional improvements to the customer support experience. These improvements include IA reconsiderations, the introduction of new features in our help center from research, and more.

We want to be sure we have a structured process for that, that will still support effective scaling as a business and resolve commonly-occurring issues.

The group running this program would also be responsible for revisiting and exploring recommendations and quick wins to continually improve the support experience and putting them on different teams' radars to consider picking up as they find themselves working on related issues.

Our hope is that there's representation from a majority of our departments in the program, so that the process of socializing improvements that could be made is efficient.

Thank you.

If you have any questions or feedback about this case study, please let me know I would greatly appreciate it.