Meta for Developers

Creation of the Application process for Compliance Review

Role
UI UX Designer

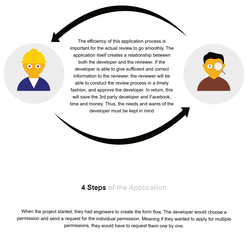
Tools
Figma

Timeline:
3.5 Years

Deliverables
User Personas, Mockups, Wireframes, Final Product

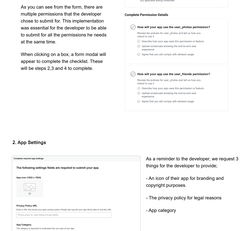
Problem Statement

Privacy of Data is of up most importance in today's age. When data gets in the wrong hands, bad things can happen. But most app's require and depend on a user's data to enhance the user's experience. To combat bad actor apps, Facebook created an application process in which 3rd party developers can acquire data permissions and features upon approval. As part of the UX team, our job was to create an efficient application process and for 3.5 years we have been working on perfecting it.



4 Steps of the Application

When the project started, they had engineers to create the form flow. The developer would choose a permission and send a request for the individual permission. Meaning if they wanted to apply for multiple permissions, they would have to request them one by one.



The issue with this implementation was the lack of detail provided by some developers. They were missing things such as setting up their profile, and providing insufficient notes and directions. Without a written statement by the developer, and if the developer did not read the docs, they would be very unprepared by our expectations of what they need to submit. Requesting permissions one by one also created back log and was inefficient. The developer would only provide one link; so the developer had to apply for the permission multiple time for multiple platforms.

In the next iterations you will see; we ask specific questions, requesting specific answers, and providing developer docs along the way.

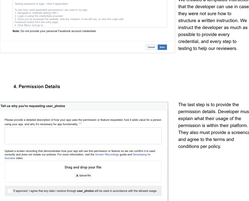
Revising the application process, we implemented a 4 step process

2. Checklist

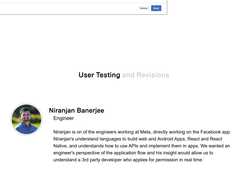
Once the permissions are chosen, a form will be presented with 2 checklist boxes. We wanted to display it as a checklist for the developer to be prepared for what they need to complete for each step.

As you can see from the form, there are multiple permissions that the developer chose to submit for. This implementation was essential for the developer to be able to submit for all the permissions he needs at the same time.

When clicking on a box, a form modal will appear to complete the checklist. There will be steps 2,3 and 4 to complete.



2. App Settings



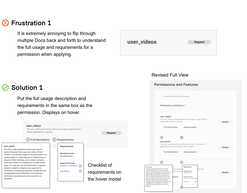
As a reminder to the developer, we request 3 things for the developer to provide;

- An icon of their app for branding and copyright purposes.

- The privacy policy for legal reasons

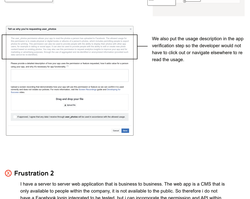
- App category

3. App Verification



If the developer has not yet provided their website, app or other platform in their settings, this modal will redirect them to do so before they can continue with the app verification process.

4. Permission Details



Once the Developer provided his platform(s), the step moves onto providing instructions.

The developer can choose which platform he wants to use the permission for. By policy, each permission on each platform must be tested. Some developers may also not need to provide

We created a templated instruction that the developer can use in case they were not sure how to structure a written instruction. We instruct the developer as much as possible to provide every condition and every step to testing to help our reviewers.

The last step is to provide the permission details. Developer must explain what their usage of the permission is within their platform. They also must provide a screencast and agree to the terms and conditions per policy.

User Testing and Revisions

Niranjan Banerjee
Engineer

Niranjan is on of the engineers working at Meta, directly working on the Facebook app. Niranjan's understand language to build web and Android Apps. React and React Native, and understands how to use APIs and implement them in apps. We wanted an engineer's perspective of the application flow and his insight would allow us to understand a 3rd party developer who applies for permission in real time.

Frustration 1

It is extremely annoying to flip through multiple Docs back and forth to understand the full usage and requirements for a permission when applying.

Solution 1

Put the full usage description and requirements in the same box as the permission. Displays on hover.

Revised Full View

We also put the usage description in the app verification step so the developer would not have to click out or navigate elsewhere to re read the usage.

Frustration 2

I have a server to server web application that is business to business. The web app is a CMS that is only available to people within the company, it is not available to the public. So therefore i do not have a Facebook login integration to be tested, but i can incorporate the permission and API within my code to generate a result in the user interface, I get continuously rejected for this because the reviewer thinks that a Facebook login is missing, and that the app can be tested for the permission.

Solution 2

After collaboration with the policy team, the UX team went ahead and made revisions to the App Verification process.

The revision includes a question at the top, asking the developer if their app includes a Facebook login. The answer is defaulted at yes, and the modal will display questions as previously shown in this case study. If the developer clicks no, the modal will change to questions asking the developer to explain how their app is server to server and to elaborate on that.

If the developer indicated their app is server to server; the permission details modal will have some slight changes to it. An input is placed for the developer to provide their github code for review.

Additionally in the screencast section, the developer is instructed to show how the Marketing API is used within their code. This in return helps the reviewer understand what to look for in their github code, thus making efficient review time.

In Conclusion

In conclusion, the revising of the application process has created a better efficiency for the entire compliance review process. The revision of the application process with the revision of the review tool has increased the accuracy of the reviewers from 60% to 99%. It has also impacted Consistency positively by increasing it from 14% to 73%. Because of these revisions, we have a better grip on eliminating bad actor apps and stopping them from having access to user data.