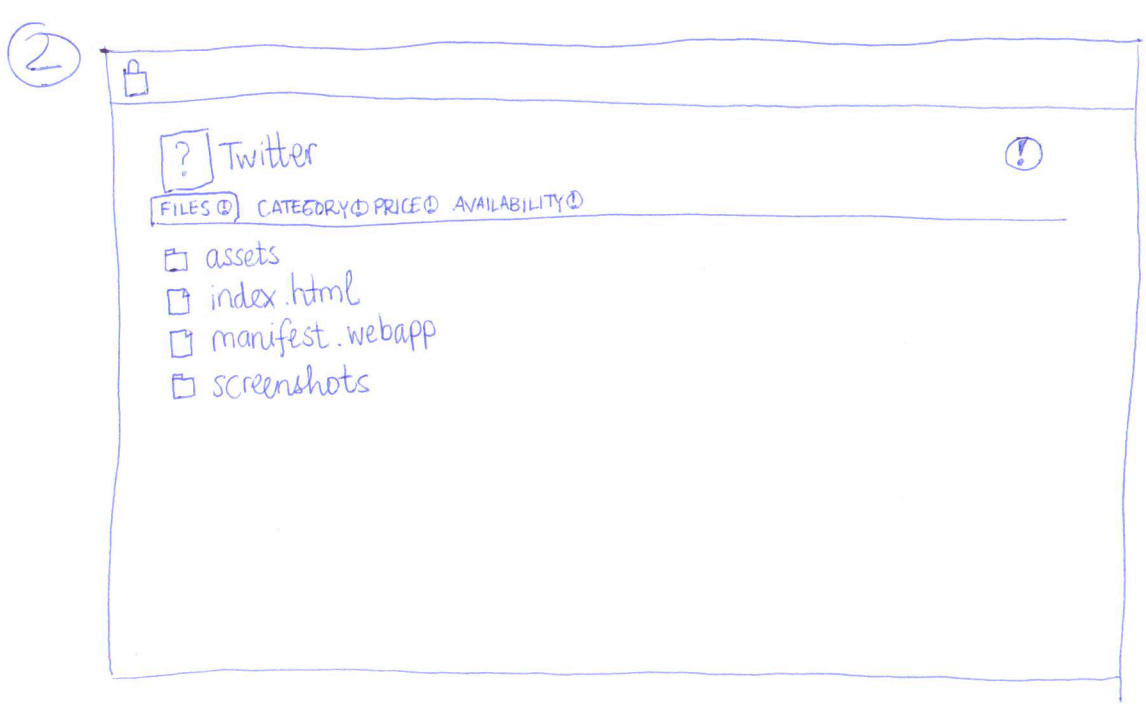# UI flow: Devhub hosts remote repositories

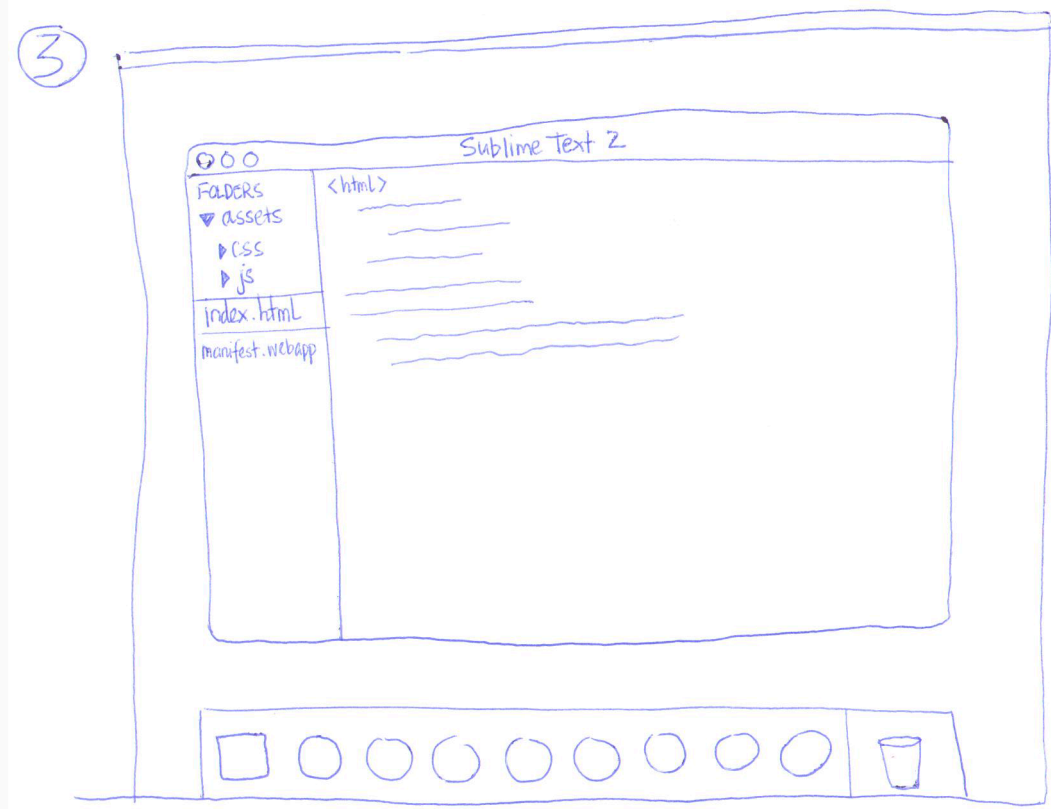User goes to Devhub and creates a new repository.

When created, a repository has the option to be initialized with starter files and directories.

User clicks on the repository that's just been created.

The repository page appears. Note how it's been populated with the starter files.
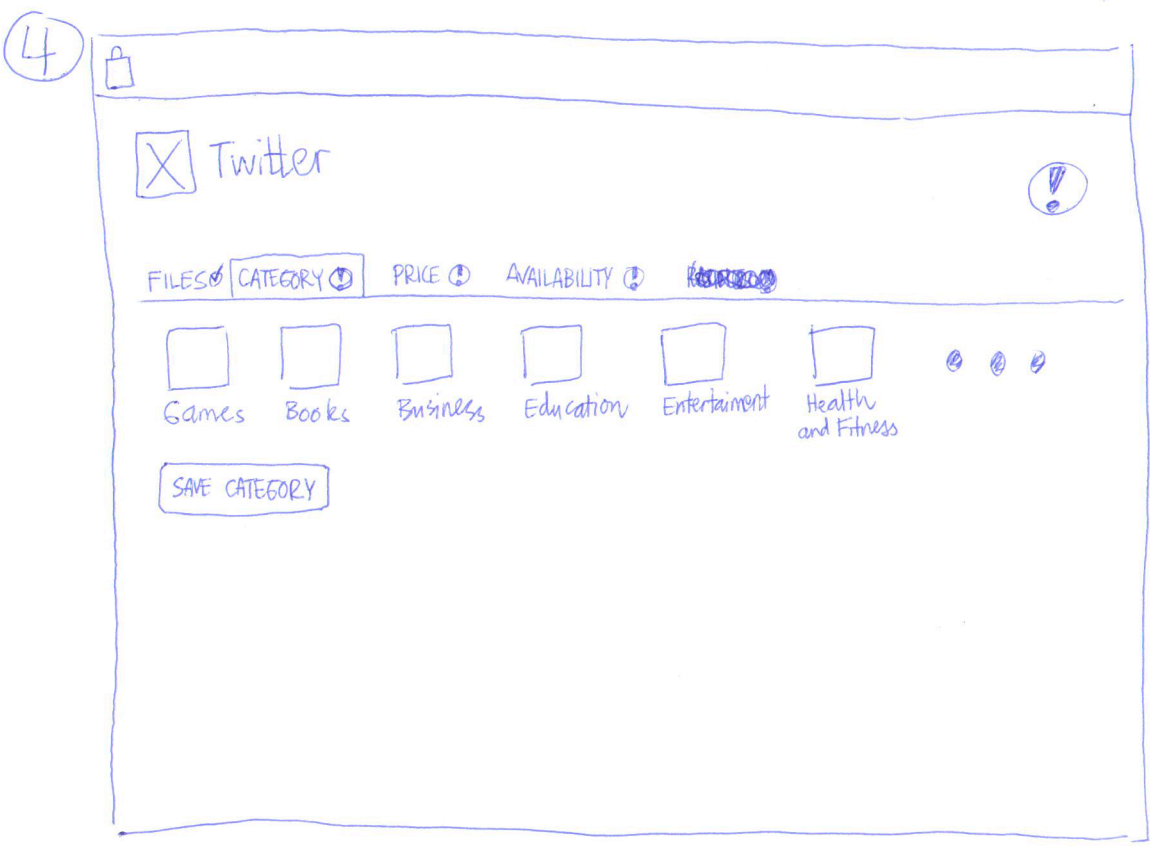
The repository should be synced to a folder that's located on the user's local machine, but this process is not shown here in the interest of brevity.
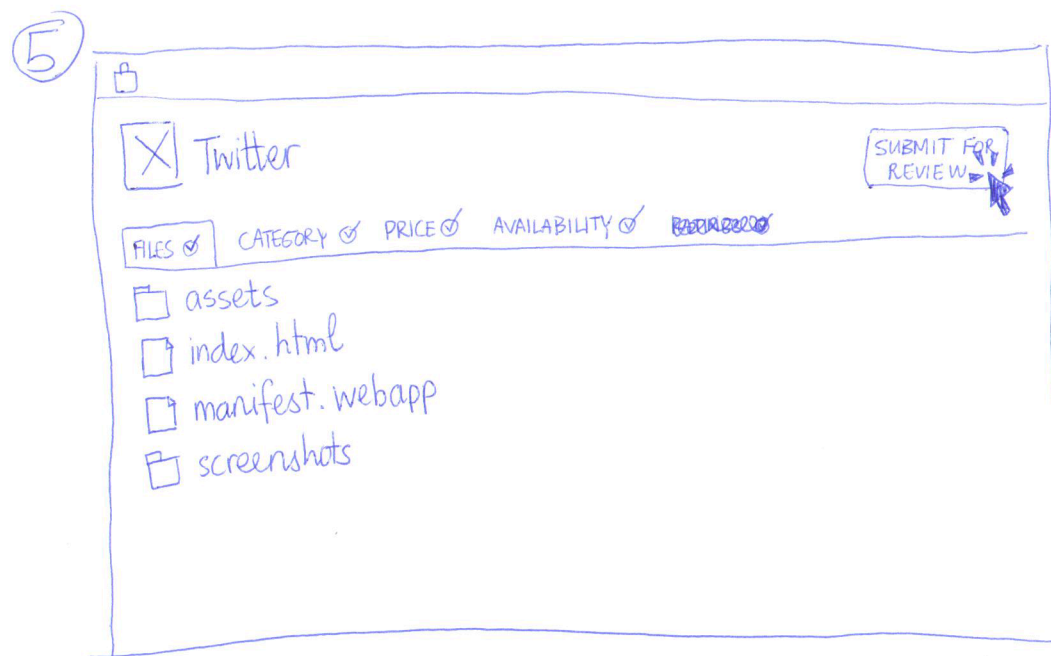

The next step happens on the user's local machine. In this step, user would go to his/her code editor or design program and start coding and creating assets.

Every time a file is saved in the working directory, it's synced to the Marketplace repository. This makes updating an app **really simple**. Open the file to update, hit "save", and the change propagates remotely.
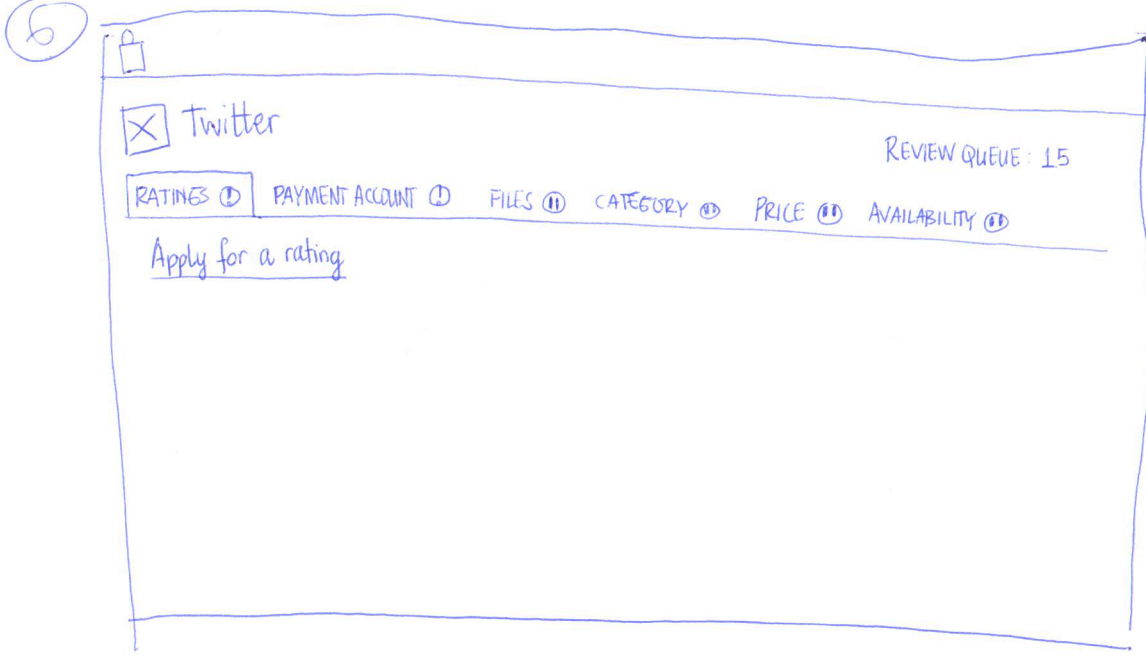
When user edits a manifest file, the code should auto-complete and be validated in order to eliminate any error that might happen on submission. This may be achieved with a .webapp plugin for text editors.

Let's say that user has finished working on the files. That section is now marked "Okay". Next, user picks a category, followed by availability and price. In this example, the category tab is shown.
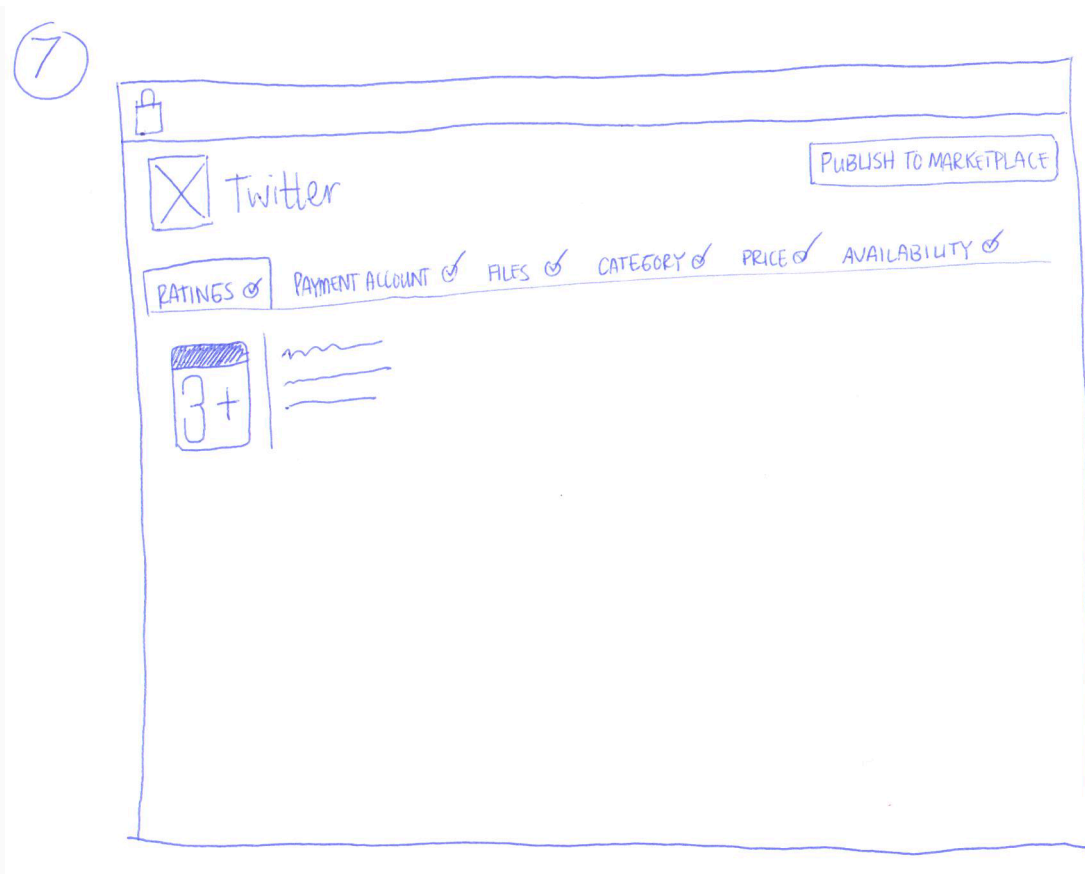

After picking category, price and availability, all the section is marked "Okay". You know what happens next. The "Submit for Review" button is enabled.

The app's been submitted. In this example, the review queue is shown. This will give user a better idea of how soon he can expect the review to be done, but it's not a critical feature to have.

Note how a (!) sign now appears under the two new tabs: Ratings and Payment Account. User will need to complete these sections. But note that Payment Account is irrelevant if the app is free.

Every section's been completed, and the app has also returned from the review process with an "Approved" status. All that's left is to click on "Publish to Marketplace" and make the app public.