

Writing Code

With Generative AI

These days AI is everywhere. I am sure you've heard or read recently "ChatGPT can write code," but what does that actually mean? What does it take to turn an idea into an application? What parts can feasibly be handled by the AI? In this paper, I will be discussing the process of Software Development and doing an experiment to see what AI can do. Now, the experiment will be an extremely simple application, not the challenges faced by software engineers in a modern commercial environment. This is basically a test to see if it can do the bare minimum. The output of each experiment was a bit much to put in this document, so I have created a GitHub repo with a README.md file with the full conversation with the AI, as well as the code it output. Each experiment is in a different branch for easy comparison.



Image generated by HotPot [hotpot.ai/art-generator] with the prompt 'AI writing code'. I have no idea why she's writing it on paper. :-)

Experiment Setup: Requirements for App

Before starting, we want to do our requirements gathering where we talk to the stakeholders and find out exactly what we are trying to build. Let's say we have a list of requirement that resulting from those meetings. We take the requirements and write them in simple plain language we can feed into an LLM:

Generate an iOS app using Swift and UIKit that has the following requirements:

- There are 3 tabs across the top of the app labeled: Log, Knives, Sharpeners
- When the app starts, the Log tab is selected.
- When the Log tab is selected, the app shows a list of log entries and an Add Log Entry button to create a new log entry. Log Entries have a created date, a knife (chosen from the knives the user has stored), and a sharpener (chosen from the sharpeners the user has stored), and the optional data from the chosen sharpener. When the Add Log Entry button is pressed, a popup screen allowing the user to enter a new Log Entry record. This popup will have buttons Save and Cancel and if Save is pressed it will save the log entry to the local device.
- When the Knives tab is selected, the app shows a list of knives and an Add Knife button to create a knife. Knives have a purchased date, a knife type, and a knife name, and a picture of the knife. When the Add Knife button is pressed, a popup screen allowing the user to enter a new Knife record. There will be the option to take a picture of the knife to save to the record. This popup will have buttons Save and Cancel and if Save is pressed it will save the knife record to the local device.
- When the Sharpeners tab is selected, the app shows a list of sharpeners and an Add Sharpener button to create a sharpener. Sharpeners have a purchased date, a sharpener type, a the ability to add 0 or more "parameters" to the sharpener. A "parameter" is a key value pair of string type. When the Add Sharpener button is pressed, a popup screen allowing the user to enter a new sharpener record. There will be the option to add "parameters". This popup will have buttons Save and Cancel and if Save is pressed it will save the knife record to the local device.
- It should look nice and be secure.

Experiment #1: Direct Requirements Input to (vanilla) ChatGPT4

https://github.com/AnnuletConsulting/SharpKnife/tree/experiment1_chatgpt4

For this experiment, we fed the above requirements into ChatGPT 4. The output did include some code, but mostly it gave instructions on how to create and edit a project in Xcode. I already know how to do that, so it didn't really help. In fact, I tried to follow the steps and I was unable to follow step 2 because storyboards are no longer used when using SwiftUI. So this test failed, but ChatGPT4 does a lot and it can be optimized for specific tasks. So I looked at the programming optimizations available and picked what looked like the best one for my needs for the next experiment.

Experiment #2: Direct Requirements Input to (optimized) ChatGPT4

https://github.com/AnnuletConsulting/SharpKnife/tree/experiment2_chatgpt4-code

I selected the code optimization available at <https://chatgpt.com/g/g-cksUvVWar-code-python-java-c-html-sql-javascript-react-web> for this experiment. There are a LOT to choose from. So many, in fact, that I wish there was an optimization to help you pick the best optimization based on your task!

The process for this experiment started out the same as before. I put in the requirements exactly as shown above. This time, I got a directory structure and some code. I started copying the code into the appropriate files in the designated locations. The output stopped and I had to click Continue a couple of times. I'm not sure if there's an output limit or something, but when it finished, I noticed not all the files had been defined, so I added an additional prompt:

In the above app, what is the content of the following files:
SceneDelegate.swift, LogEntry.swift, Knife.swift,
Sharpener.swift, LogEntryCell.swift, KnifeCell.swift,
SharpenerCell.swift, and DataStorage.swift?

It responded with the code for the files. I copied all the code into the correct files and made sure they were all in the correct Target for the project. At that point I had 2 build errors on the Sharpener file. I added an additional prompt:

```
How do I fix the error: Cannot automatically synthesize  
'Decodable' because '[(String, String)]' does not conform to  
'Decodable'
```

Notice I did NOT mention that the error was in the Sharpener file. Therefore, I was surprised that the output knew which file had the error and returned

```
The error occurs because Swift's automatic synthesis of the  
'Decodable' protocol for a struct fails when one of the  
properties is a type that does not conform to 'Decodable'. In  
this case, '[(String, String)]' does not automatically conform to  
'Decodable'.
```

To resolve this, you need to manually implement the 'Decodable' conformance for your 'Sharpener' struct. Here's how you can do it:

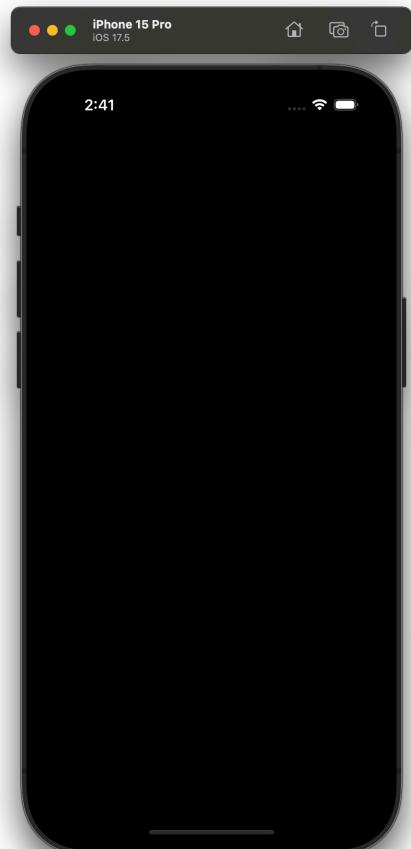
Then it output new code for the Sharpener.swift file, which I copied over the previous output. This fixed the errors! I excitedly hit BUILD and it built the app with no errors. Awesome! So I fired up an iPhone Emulator and ran the app. Hmm... a blank screen (screenshot of emulator on right) is not what I was expecting.

Thankfully we have our helpful AI to ask! My next prompt was:

```
This code compiles, but when I run it on an  
iPhone, I just get a blank screen. Why?
```

The full output is in the branch README but basically it said:

Getting a blank screen on your iPhone when running



the app can be caused by a few different issues. Here are some common reasons and how to fix them:

1. **Window and Root View Controller Setup:** Ensure that the window and root view controller are correctly set up in the `SceneDelegate`.
2. **Missing Views or Constraints:** Ensure that your views and constraints are correctly set up so that the UI elements are displayed properly.
3. **Data Source and Delegate Implementation:** Ensure that your table view's data source and delegate methods are correctly implemented and that data is being loaded.

Here's a checklist and code snippets to help you debug and fix the issue

followed by the same code it had given us for the `SceneDelegate` and `ViewControllers` in previous steps. There were no changes to be made based on what it suggested. Without some knowledge of coding/debugging, there isn't anywhere else to go from here. You could possibly ask this same question in other ways and possibly get a usable answer, but clearly we can't just dump in our requirement and get a working app output!

Further Considerations

What we were doing above, developing new software from the ground up, is often referred to as Greenfield development. Most development work is maintenance or brownfield. According to ChatGPT4:

Estimates suggest that around 70-80% of software development jobs involve maintenance and brownfield development, with only 20-30% being greenfield projects. ([The Tech Report](#)) ([Synoptek](#))

So perhaps a more realistic experiment would be to do some maintenance coding, and let's help the AI out a bit and do what we do best, break down big problems into smaller ones.

Experiment #3: Breaking It Down Into Smaller Tasks

https://github.com/AnnuletConsulting/SharpKnife/tree/experiment3_task1
https://github.com/AnnuletConsulting/SharpKnife/tree/experiment3_tasks2-5

In most software development environments, project requirements are broken down into smaller tasks that can easily be estimated and managed. If using the popular project management software, Jira, one might have a project that has several Epics, which are broken down into Stories, which can be broken down into Tasks and even Subtasks. Often there might be unknowns, so you have a timeboxed Spike to do some research. In this experiment we will take an existing project that needs some work and see if the AI can make the small changes we ask to the existing codebase. And we just happen to have created one in the last experiment, how convenient!

Task 1: App Shows a Blank Screen

I run the app, and sure enough it builds and runs and shows a blank screen. Using my knowledge of UIKit in Swift, I look at the AppDelegate and the SceneDelegate code. I notice immediately that they are almost identical. I know that generally you want to avoid code duplication. So I ask the AI with the following prompt:

Why are SceneDelegate and App Delegate both creating Tab Bars?

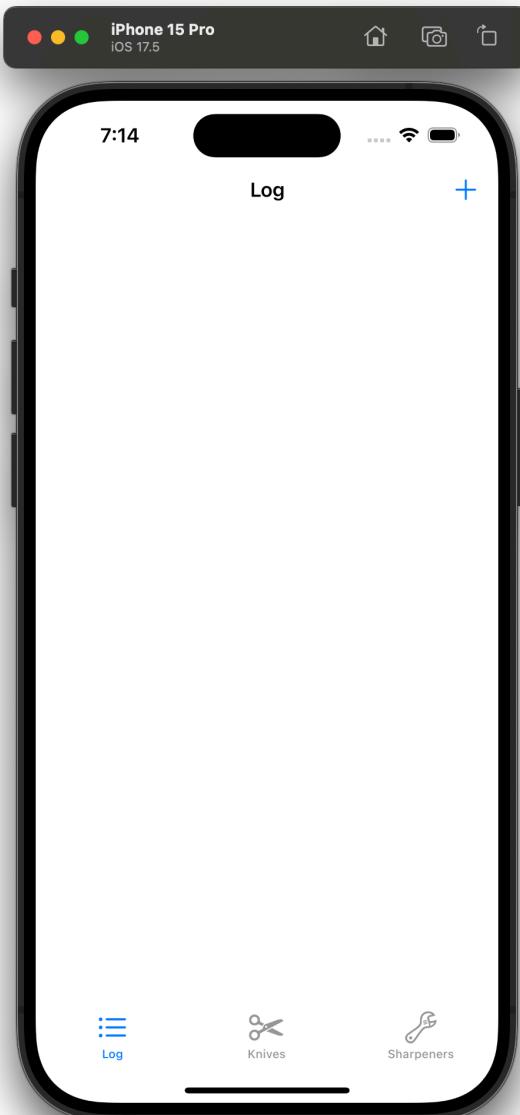
to which it replied

You're right. The SceneDelegate and AppDelegate both seem to be setting up the window and root view controller, which can cause confusion. In iOS 13 and later, SceneDelegate is responsible for managing the app's UI lifecycle, while AppDelegate handles application-level events.

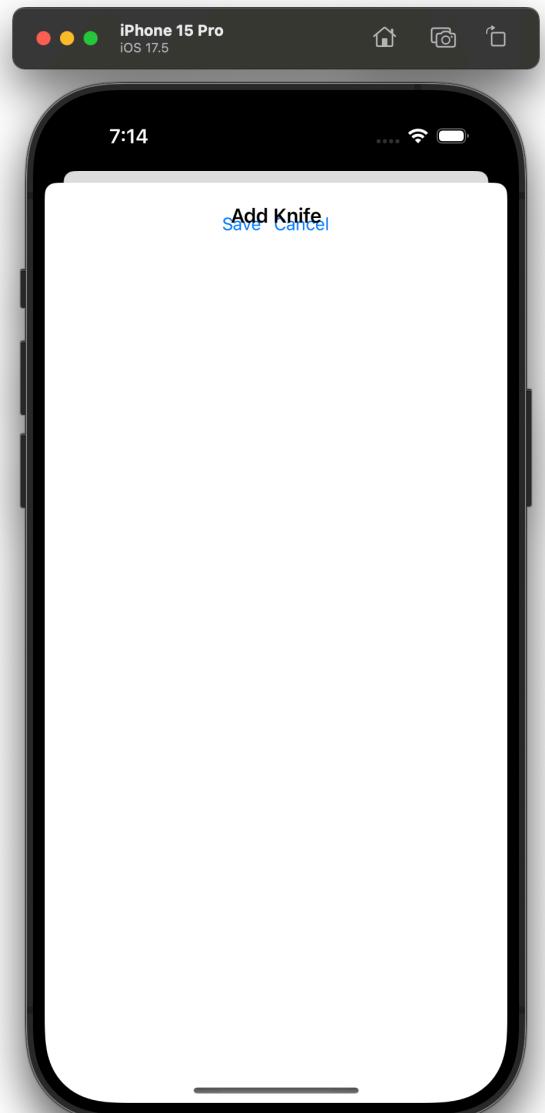
Let's clean this up by ensuring that SceneDelegate is correctly being used to set up the window and root view controller. Here's how you can ensure SceneDelegate is correctly configured:

The reply, which you can see in its entirety in the README file, included new code from AppDelegate, as well as data to add to the Info.plist. After making both of those changes I ran it again and there's an app there instead of the blank screen! There is an app, but it's not perfect. We fixed one bug and found several new

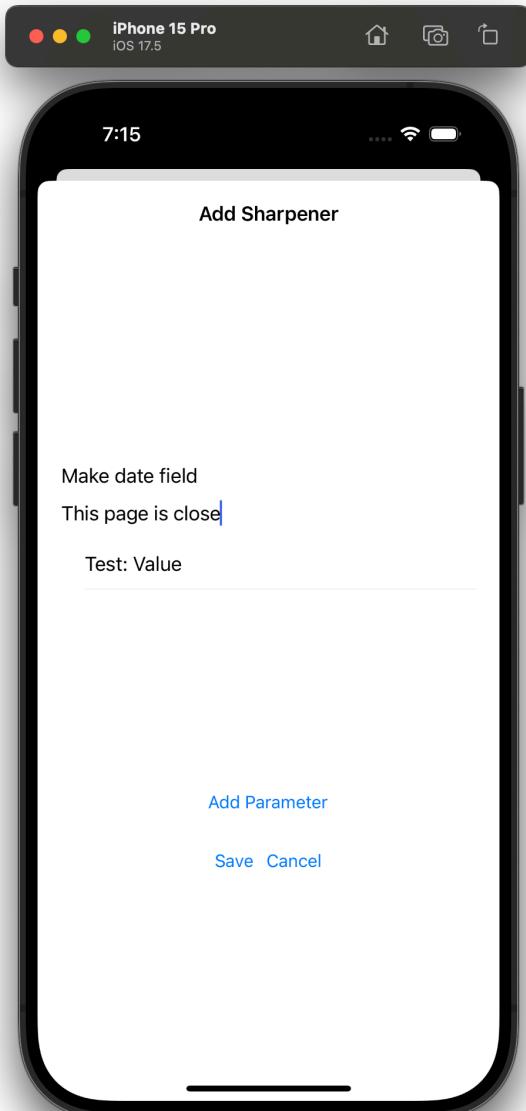
ones! So this is turning out to be a realistic test after all. Here are the screenshots of the app after making the suggested changes.



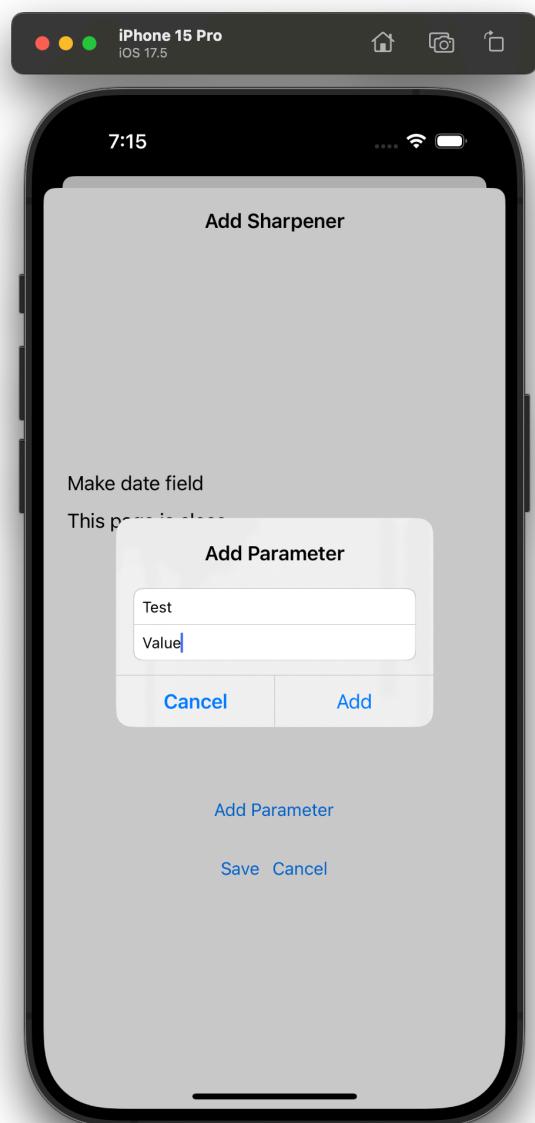
Ok, the log page looks good, it's empty of course, but that is to be expected. Same with the Knives and Sharpeners page also.



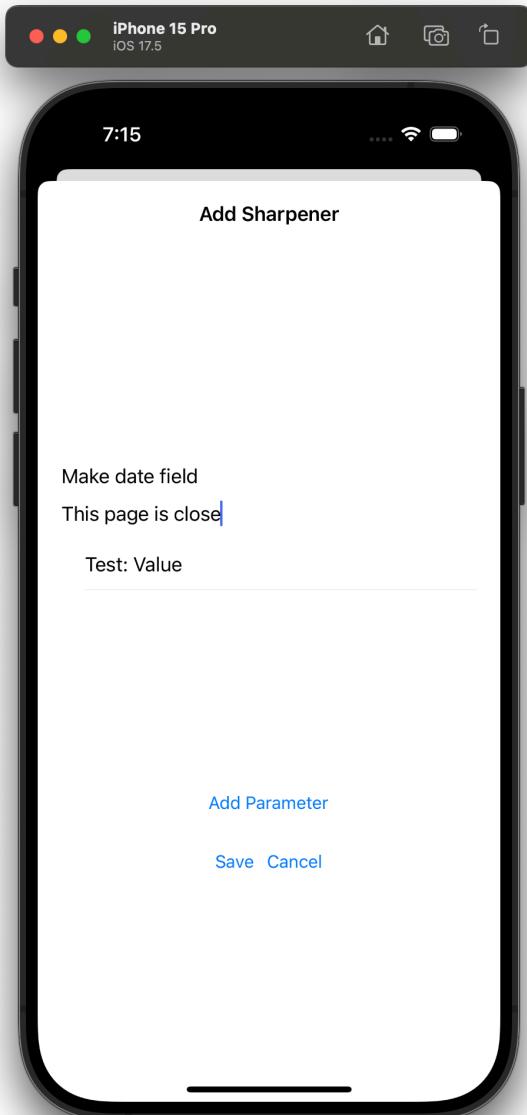
When opening the Add Knife popup, we get some error in the console and see the screen isn't rendering properly. Ok, let's make a Task for that and we'll come back to it after we've done our impromptu QA.



In the Add sharpener page, I can type non-Dates in the date field... that could be done better, so we'll add a new Task to our list. But there's a name field, and the Cancel button closes the window.



The Add Parameter button button works.



And clicking Add attaches it to the Sharpener you are adding.

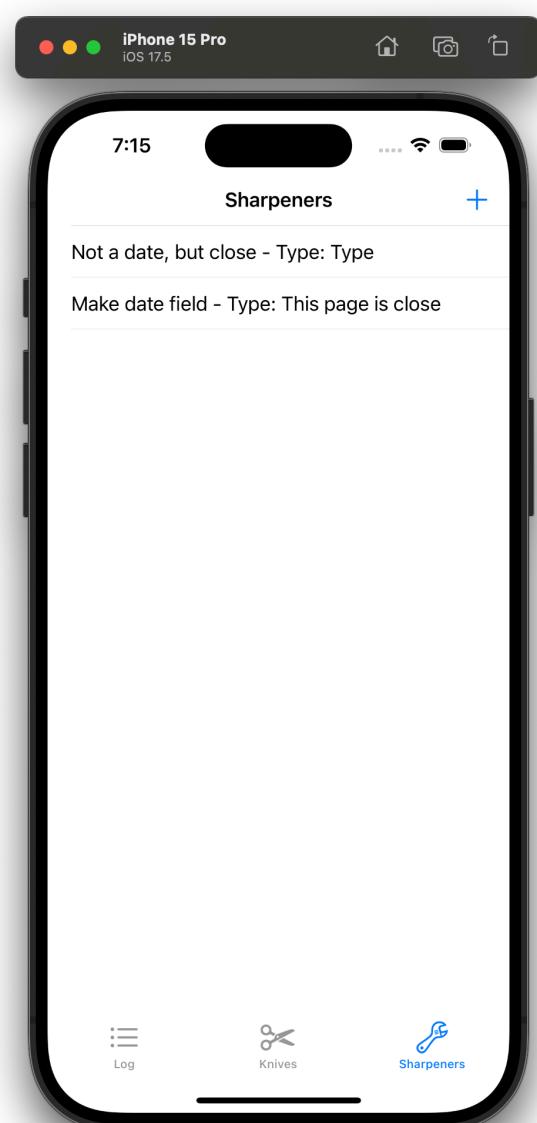
Make date field

This page is close

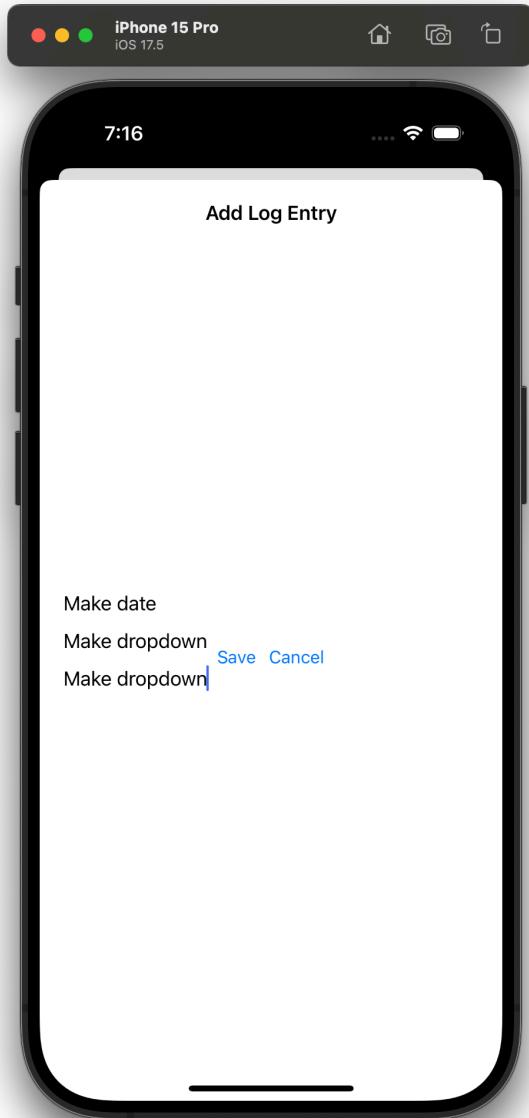
Test: Value

Add Parameter

Save Cancel



Clicking Save, saves the sharpener and you can add more than one to see them in a List.



The Add Log Entry page needs a date input makeover as well, so we'll change our earlier task to include all date inputs. Once we do it for one, we can reuse that code for all of them.

We also need to make the two text input fields dropdown lists that allow the user to choose from the ones they have saved.

And finally those buttons need moved down.

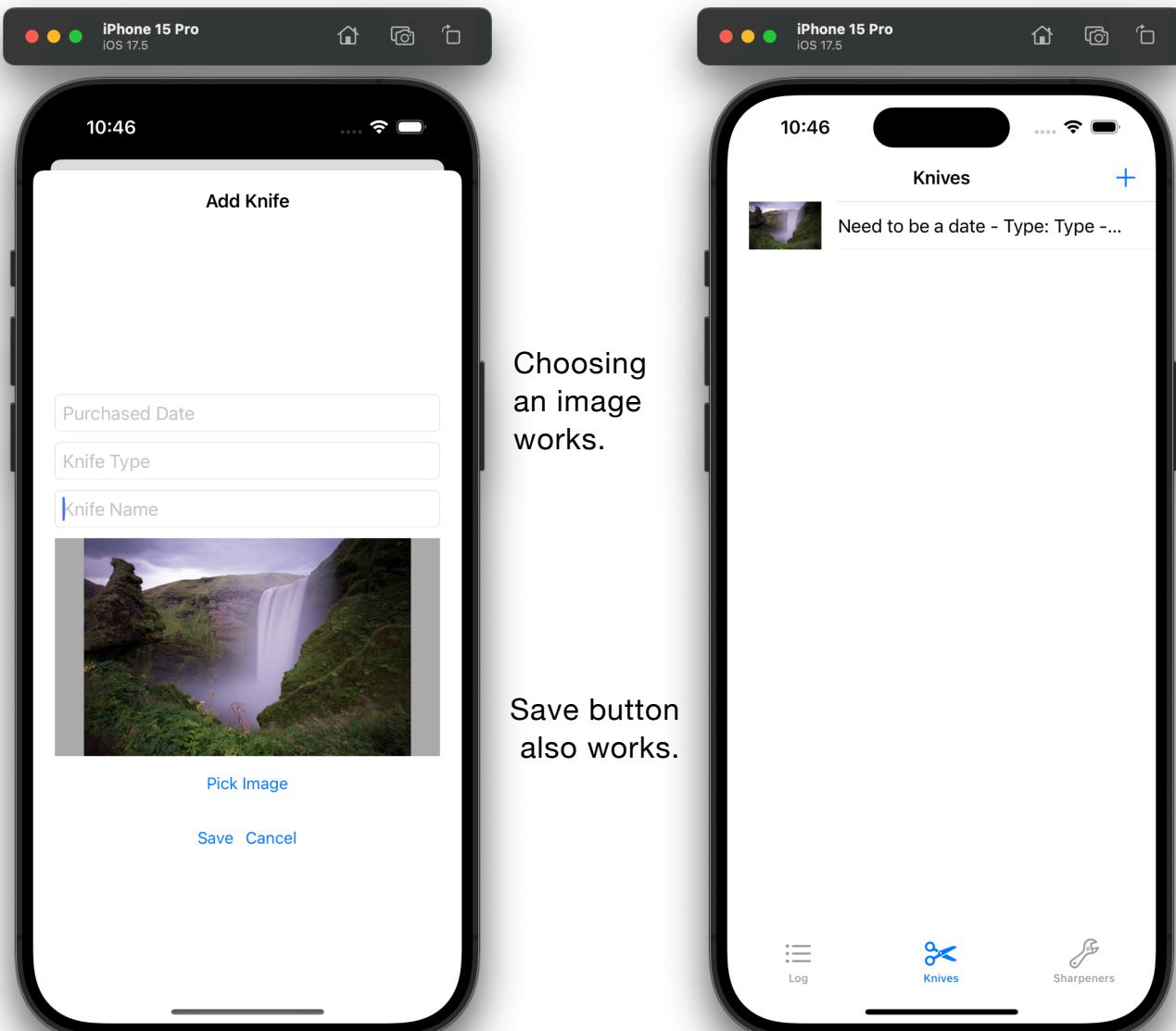
That's all the problems I see so far, so lets work on these new tasks now.

Task 2: Fix Error on Add Knife Screen

Ok, let's see if it can fix the error without me using any coding-fu and just plugging the error into the prompt:

The app is running now, but when I click on the Knives tab and then the + icon, the Add Knives screen is blank and I get the following error in the console: [[Error copied from Console, see README for full chat conversation.]]

It replied with no code for the AddKnifeViewController and an explanation of the changes it made. They all make sense, so I pasted the code over the old class and tried again. Hey, look at that!



Task 3: Improve Input for Date Fields

Now we want to see if it can change the Dates to a Date Picker instead of Text. In SwiftUI this is a UIDatePicker but I'm just going to say date picker in the prompt in case it has other ideas.

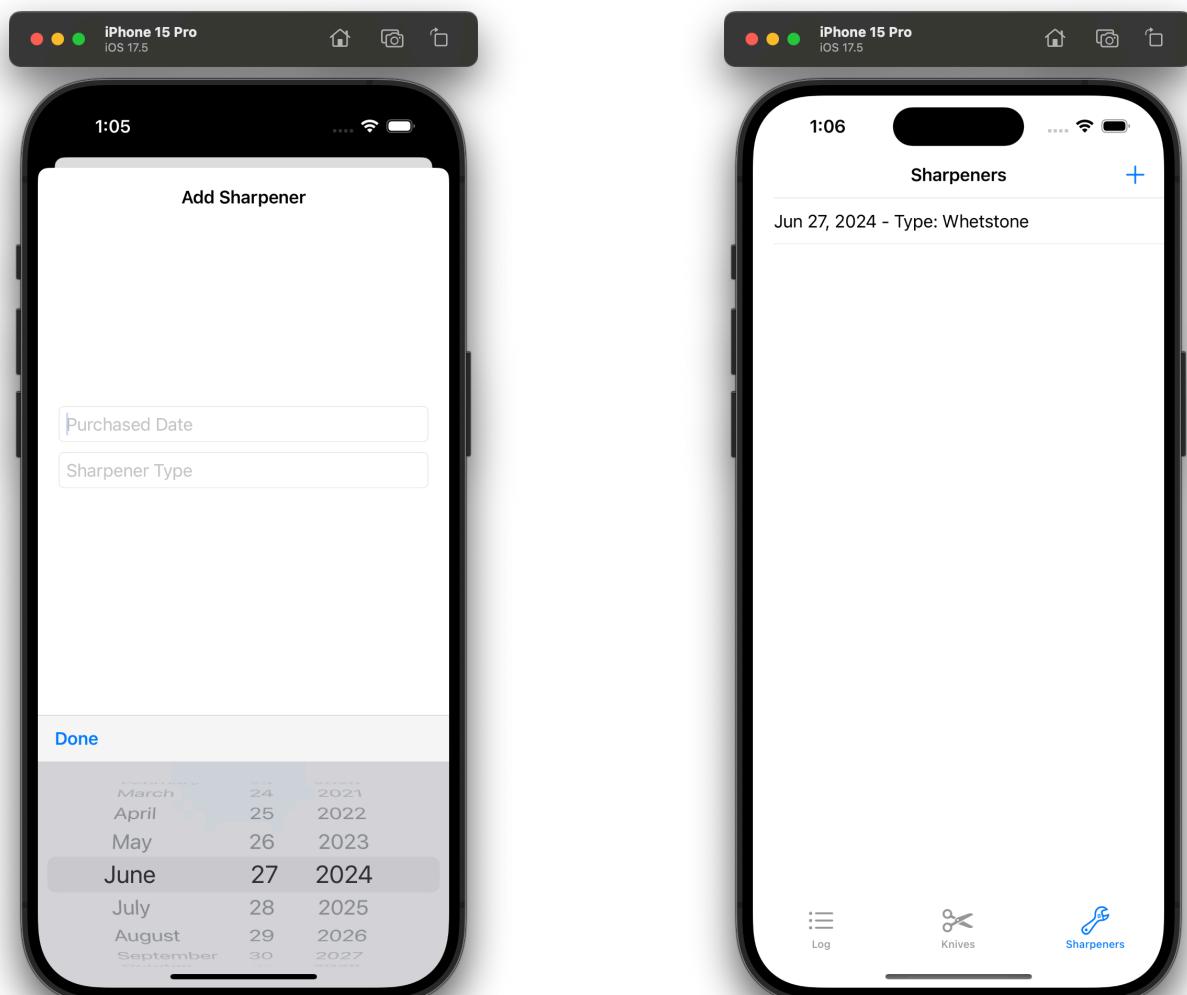
In the AddSharpenerViewController the Purchased Date is a Text Input. Change that to use a date picker instead.

It replied with

Sure! We can replace the text field for the "Purchased Date" with a UIDatePicker. We'll also add a toolbar with a "Done" button to dismiss the date picker once a date is selected.

Here's the updated AddSharpenerViewController.swift:

I copied the code it output and added it to the repo and restarted. Much better!



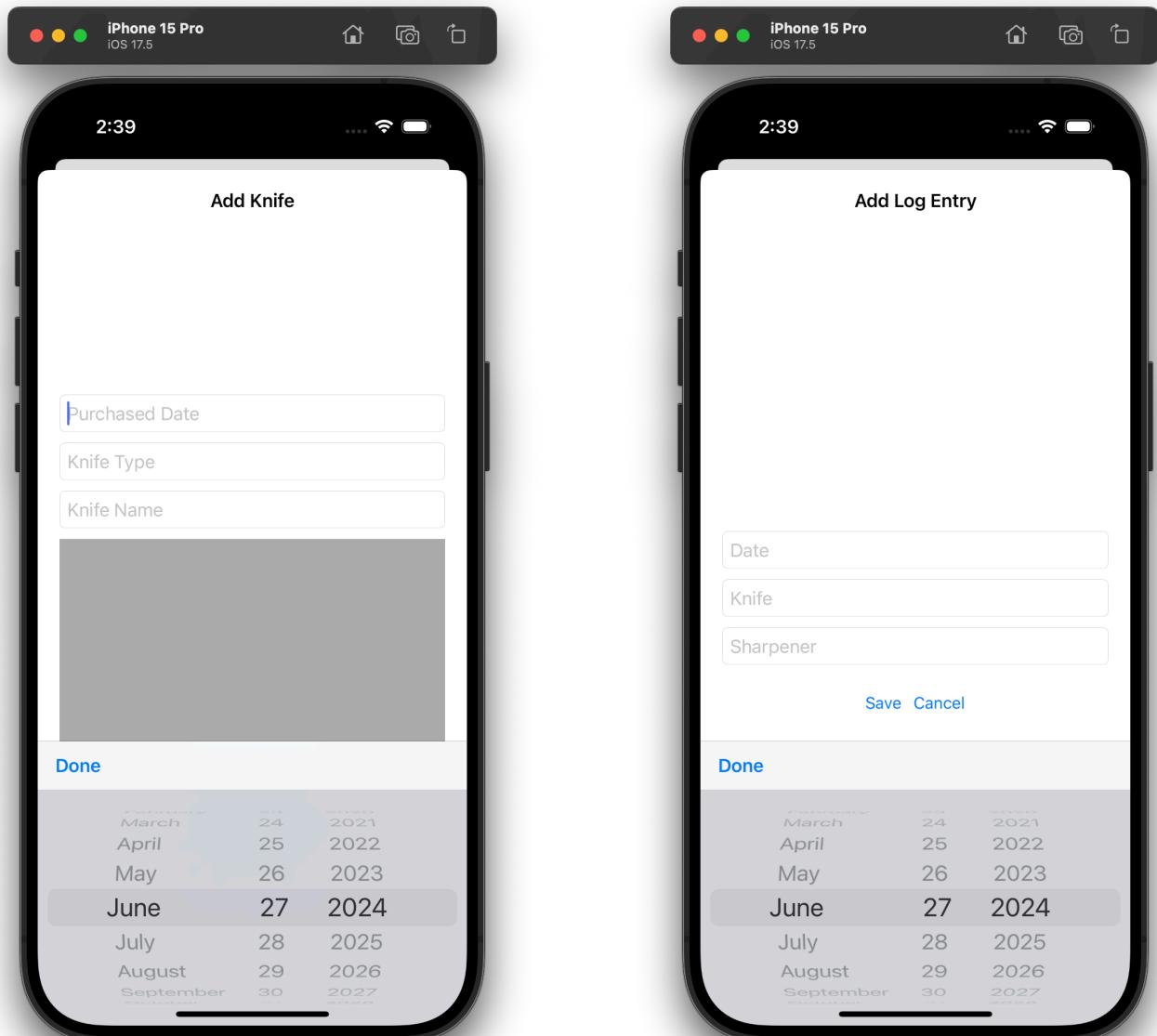
Ok, let's have it do the other two fields as well.

Please make the same change to the date fields in AddKnifeViewController and AddLogEntryViewController and show me that code.

It replied with

Sure! Below are the updated versions of AddKnifeViewController and AddLogEntryViewController to use a UIDatePicker for the date fields.

and updated code for both classes, which I changed in the repo. These now have DatePickers like the other ones



Task 4: Log Entry Page Should Have Dropdown for Knives and Sharpeners

The prompt I'm using for this task is

On the AddLogEntryViewController, change the Knife field from a text input to a dropdown list that contains all the Knives we have saved. Also change the Sharpener field from a text input to a dropdown list, this one containing all the Sharpeners we have saved. Output the code.

It output the code and at the end said "You might need to update the DataStorage methods to properly load and save the Knife and Sharpener objects." But I tried it as written and I get an indexOutOfRange error because it doesn't appear to load the knives, so I'm going to ask it to fix that.

Update the DataStorage methods to properly load and save the Knife and Sharpener objects and output the code.

It output the new code for DataStorage, but it added the models to the end. Since we already have them defined in the Models group, it throws a compilation error. I just deleted them from the DataStorage file and then it would build again, but it still isn't loading the data into the lists.

It still is not loading the data in the dropdown lists. Why?

Again, new code for DataStorage, the same as before but some log statements to help debug, and new code for AddLogEntryViewController. It's loading 0 records, and I noticed that they are not persisting between app restarts, which suggests they're only in the memory, not persistent storage. Looking at the code, I can see that it is not calling DataStore utility, only appending to the array.

In the KnivesViewController, the didSaveKnives extension function does not call the saveKnives function in DataStorage. Also, in the LogViewViewController, the didSaveLogEntry extension function does not call the saveLogEntry function in DataStorage. Additionally, in the SharpenersViewViewController, the didSaveSharpener extension function does not call the saveSharpeners function in DataStorage. Update those three Controllers and output the code.

It output the code for the 3 ViewControllers as asked. I pasted them into Xcode and ran it. I created a Sharpener and then closed and reopened the app. It is still there. I created a knife and when I saved I got an error and it is not there after restarting the app.

I saved a Sharpener and that worked, however, when I attempted to save a Knife after that, it gave the following error.
CFPrefsPlistSource: Attempting to store >= 4194304 bytes of data in CFPrefs/NSUserDefaults on this platform is invalid. This is a bug in SharpKnife or a library it uses.

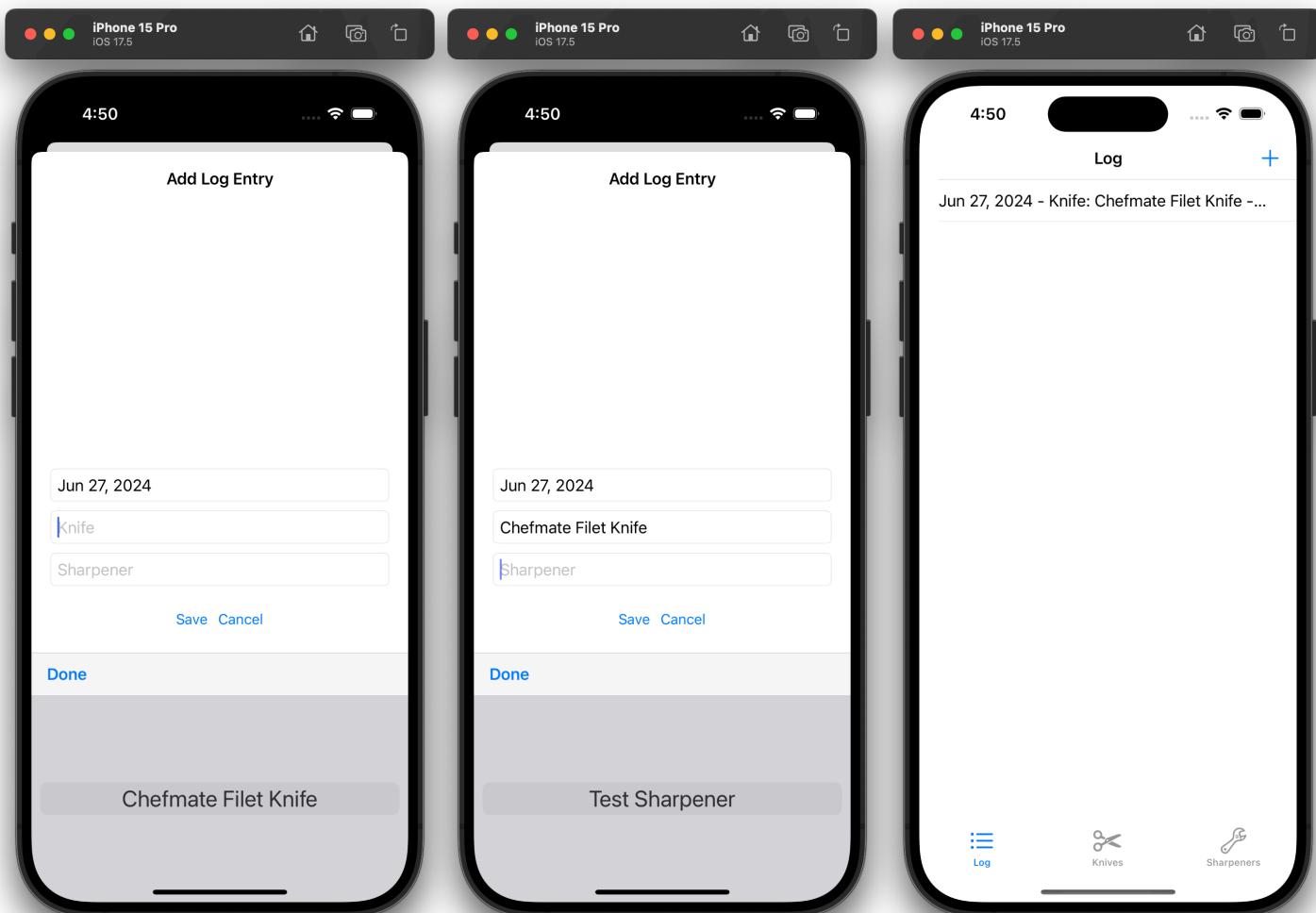
Description of keys being set:

knivesKey: data value, size: 6143815

Description of keys already present:

sharpenersKey: data value, size: 88

It responded that trying to save the whole image was too big and it made some changes to the Knife and KnifeCell classes to save only a URL for the image instead of the image itself. Now knives are being persisted with the images loading from the url. With the data being saved between restarts, the drop-down are now loading, which means we can save out first log entry.

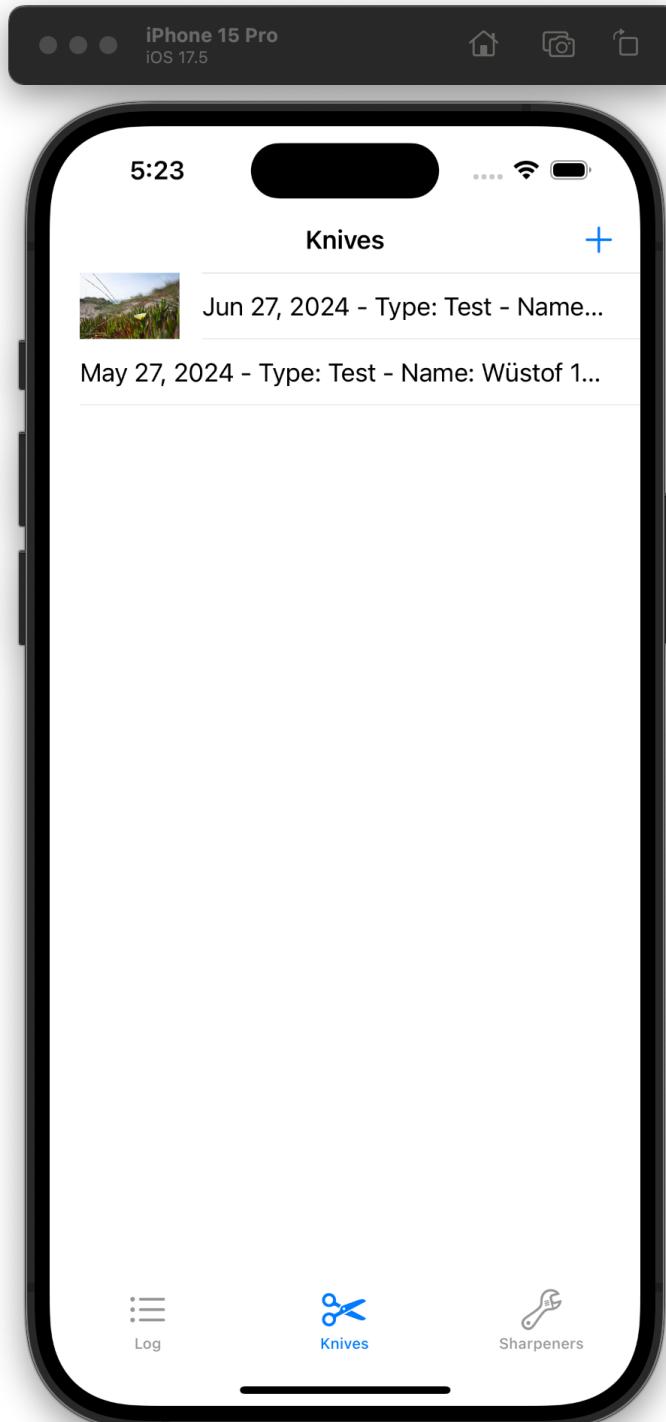


Task 5: Fix Must Have Image Issue

In my testing, I noticed that it wouldn't let me save if I didn't pick an image. I'd like to make images optional, so I prompted the AI:

It's saving now, but it requires an image to be selected to save.
Alter the code so it will still save the knife even if I do not select an image.

It gave new code for AddKnifeViewController, which worked when pasted it into Xcode.



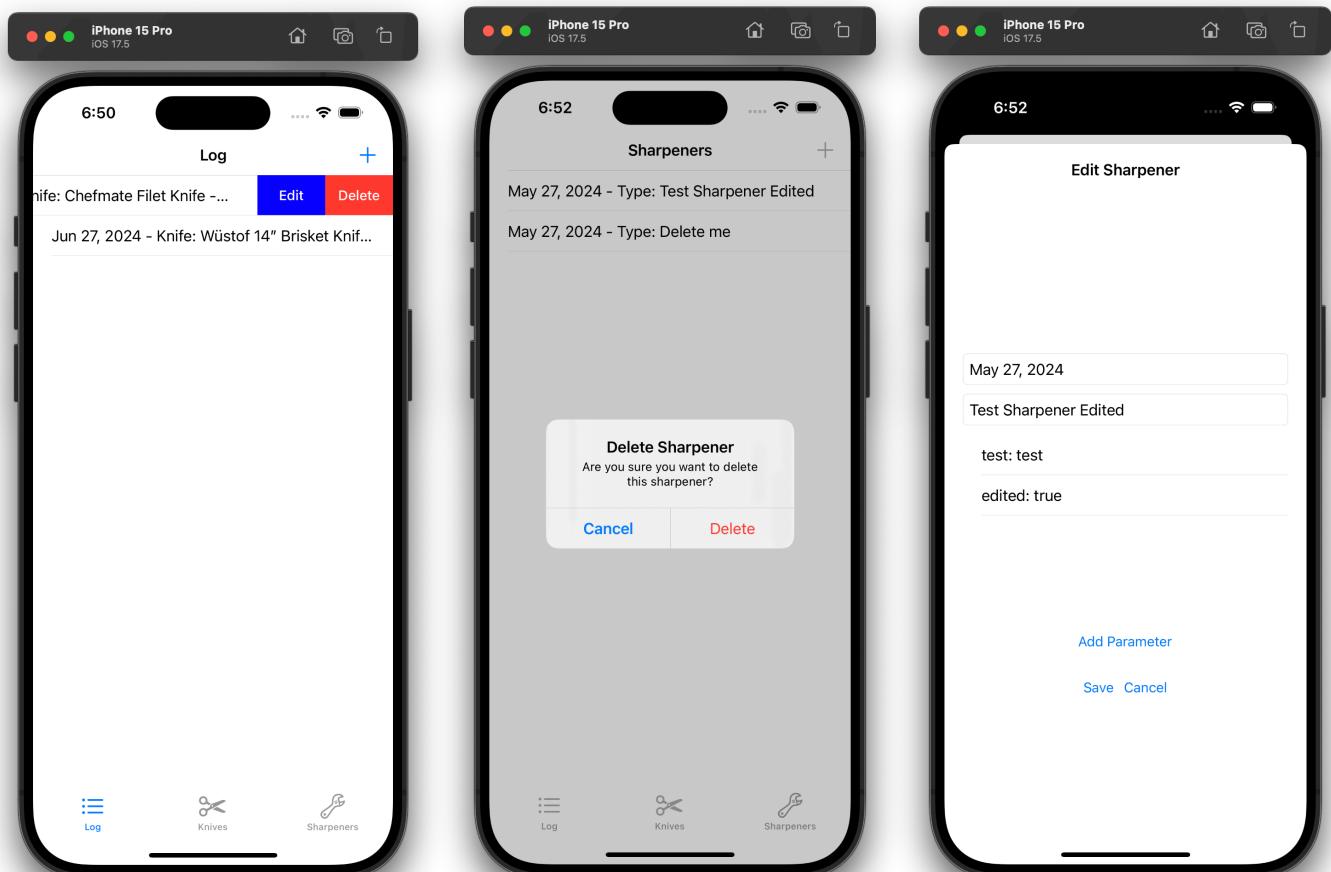
Experiment #4: Adding Features

<https://github.com/AnnuletConsulting/SharpKnife/tree/experiment4>

The app is doing what we originally asked for with our requirements. It took a bit more work than we hoped, but the AI is doing an admirable job so far. But now looking at it, it is obvious we left out some obvious features. Let's add the ability to Edit and Delete the three types of records.

In the LogViewController, KnivesViewController, and SharpenersViewController, alter the tableView to allow the user to swipe left on a list item to expose a delete button, that when pressed will ask for confirmation to delete the record from the device. Also make those tableViews show an Edit button when swiped right on a list item that when clicked will show an Edit page for the item. Also show the changes required to DataStorage to support these changes.

At this point it started getting slow and the responses would stop in the middle of the code output and never resume. I finally tried paying \$20 to upgrade from the free account to the Plus account and suddenly it started working again, so I assume I'd maxed out the free usage amounts and was getting throttled. After I paid, I was able to get the file output and put them in Xcode and run it. It's not EXACTLY what I asked for, but close enough so I'm going with it.



Experiment #5: Changing Requirements

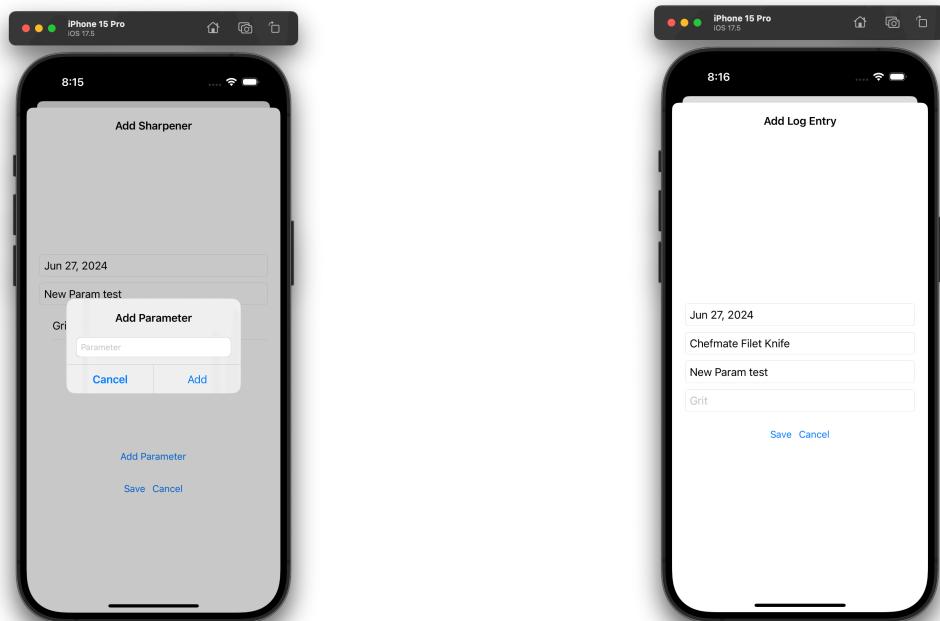
<https://github.com/AnnuletConsulting/SharpKnife/tree/experiment5>

So we've seen how it can do regarding adding to the original requirements, what about if we change the original requirements. Will that lead to confusion or will it always work from the latest version of the requirements? Originally when I did parameters on the sharpener, I thought different sharpeners might have different attributes, like grit. But others wouldn't so it was more informational. But what if the sharpener is adjustable? What if the angles can be changed and it has different grit pads? We don't want to have a sharpener record for every configuration of it. So lets move the parameter values to the Log record and only have the key in the sharpener record. While we're add it, lets add the ability to delete parameters if we make a mistake.

Make the following changes and output all code that was altered:

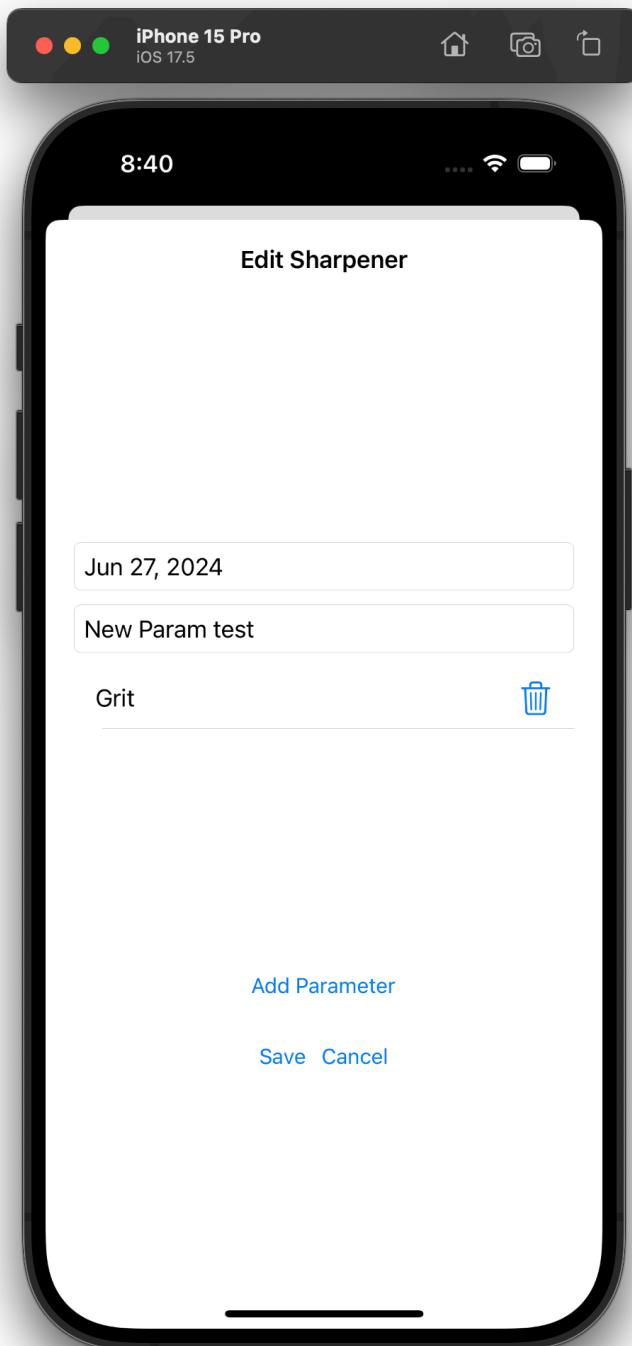
- Change the Sharpener model so the Parameters are strings instead of key value pairs.
- Update Add Parameter popup on the Add/Edit Sharpener page to reflect the change to the model.
- Add a trash can icon to the right of each parameter on the Add/Edit Sharpener page that will delete the parameter when clicked.
- Update the Add/Edit Log Entry page so that when a sharpener is selected with the dropdown, it dynamically creates text input fields for each of the parameters for that sharpener.

It made the changes, but didn't show all the files I needed. I had to ask twice more for files it forgot to tell me it changed. Once I coaxed all the changed files out of it, I ran the new code. The delete buttons aren't showing up, and it isn't saving the parameter values in the Log Entry.



When I save a Log Entry it is not saving the values I entered in the parameter fields it is dynamically creating. It should save those and repopulate them when Edit is tapped. Also, the delete buttons for the Parameters are not showing on the Add/Edit Sharpener page, please fix that issue.

The Parameters are now saving and being shown when the Log Entry is saved and then edited. The delete buttons are still not showing. There was an error in the console, so I asked the AI if that error could be causing the delete button not to show. It said it probably was and output some new code for a few classes. I pasted them in and ran the app and now we have a delete icon!



Experiment #6: Data Manipulation

<https://github.com/AnnuletConsulting/SharpKnife/tree/experiment6>

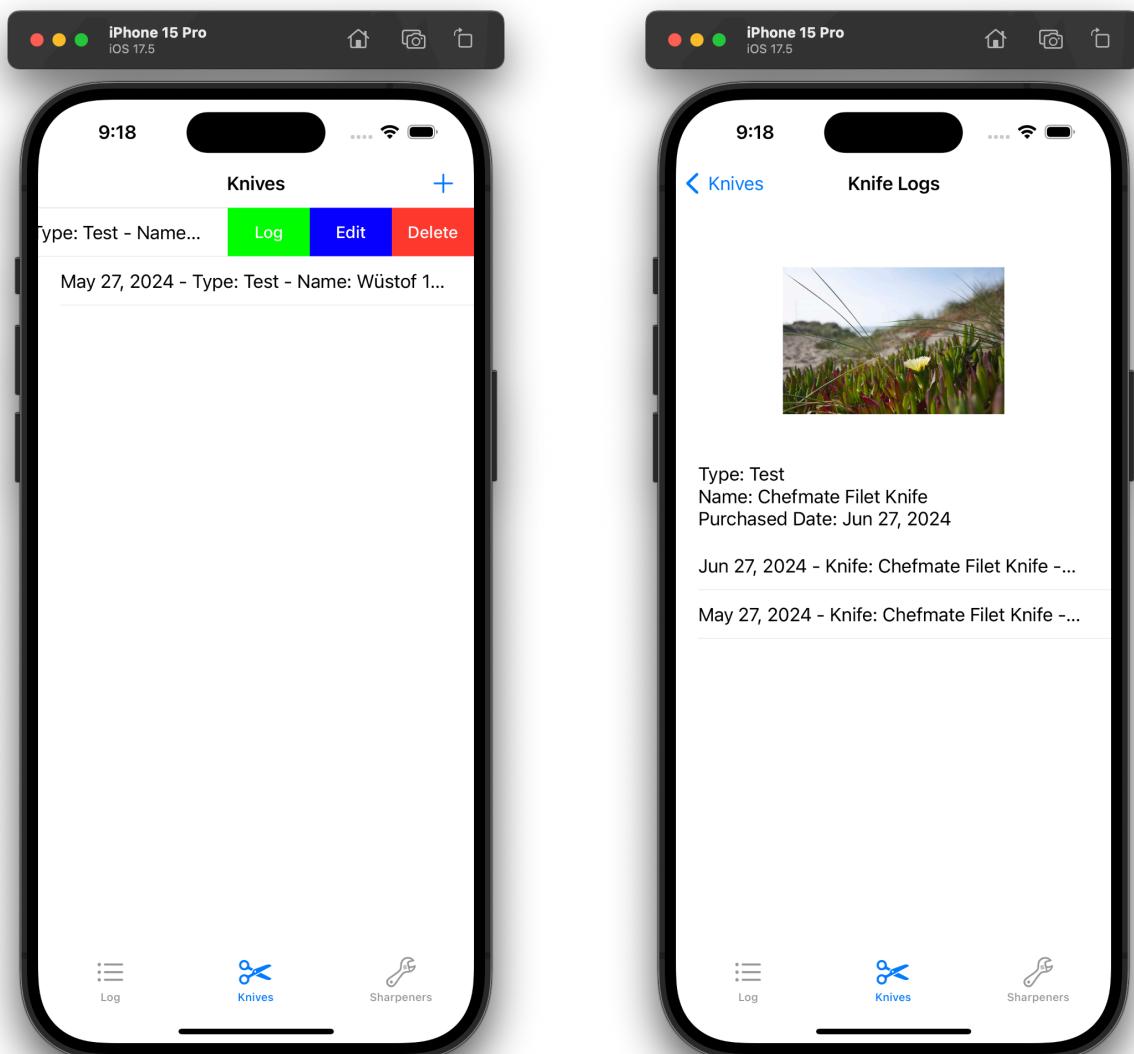
We've done simple CRUD (Create, Read, Update, Delete) data operations so far. Now we will attempt some very simple reporting. On the Knives table, we want to add an option to show Logs for just that knife.

On the KnivesViewController, next to the Edit and Delete buttons on each knife, add a Log button. When that button is clicked, it will open a view showing a header at the top with the Type and Name and Purchase Date and Image (if one exists) and below it will show a table of the Log Entries for that knife.

It gave me new code but the build failed.

The build failed with error: Cannot assign value of type 'KnivesViewController' to type '(any KnifeCellDelegate)?'

This code worked, and we have our logs!



Experiment #7: Manual Changes

<https://github.com/AnnuletConsulting/SharpKnife/tree/experiment7>

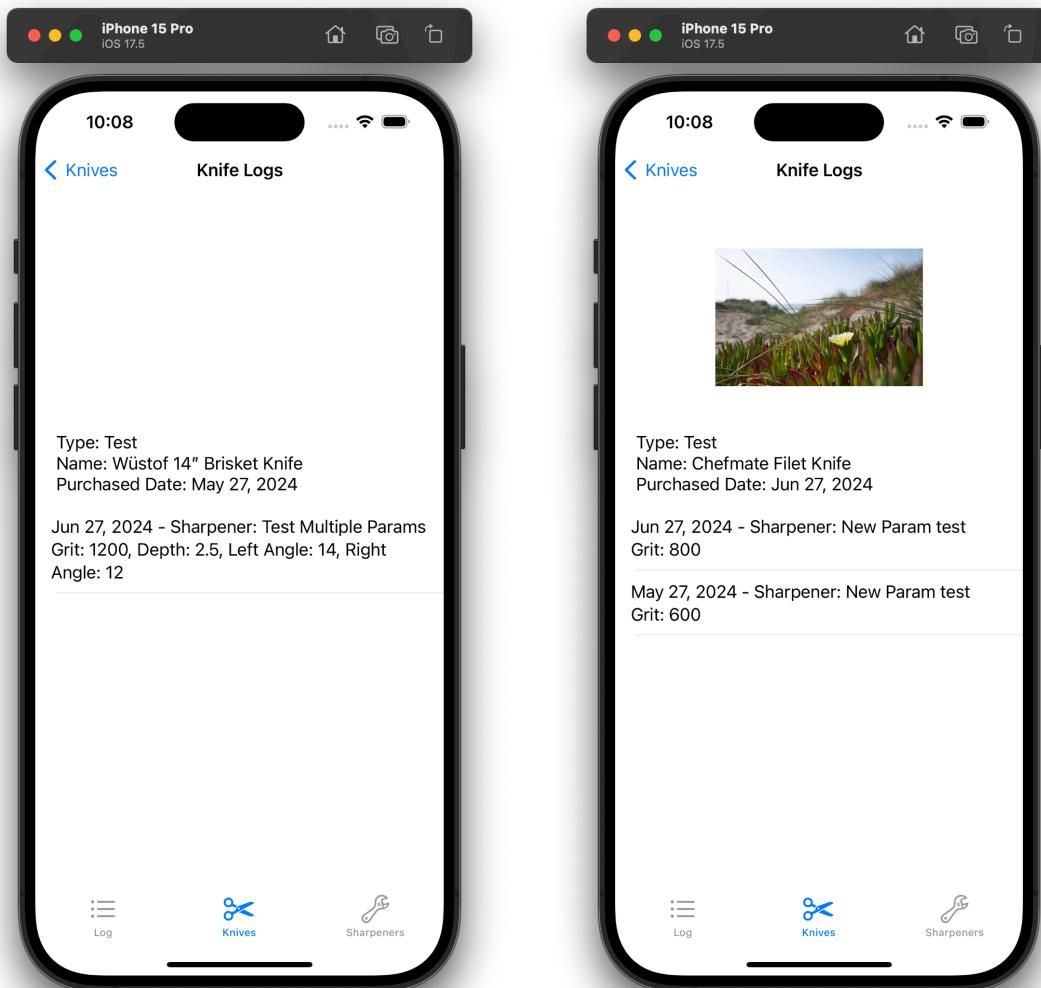
So far, the AI has been making all the changes, what if there is a change that we are having trouble getting it to understand, and we want to make changes to the code by hand and then get it to make changes to the new code. I made some changes to the Log View from the previous experiment and sent them to ChatGPT. See the repo README for the full details. After uploading the code changes, I asked it to

Change the LogEntryDetailCell so that it displays multiline text.

It did.

Now have the LogEntryDetailCell pull in the Sharpener referenced in the LogEntry and match the parameter names to the values stored in the log entry and display them in front of the corresponding value.

I made some more changes and uploaded them. It cleaned up the code a bit and output it back to me. Nice!



Experiment #8: Writing Unit Tests

<https://github.com/AnnuletConsulting/SharpKnife/tree/experiment8>

We have a working app, but we need some test cases to make sure it keeps working as we make changes.

Generate some test code for the iOS app we created,

It output some code and instructions on how to add the tests to the project in Xcode. I followed them and ran the tests. The DataStorage tests passed but the UITests failed. I gave it the errors and it made code changes, but wasn't able to determine the problem. I looked at the error and it was obvious the problem was the test wasn't changing to the knives tab before attempting to add a knife.

Given the below error, I think the code for testAddKnife is looking on the Log Entry page. The test should switch to the Knives tab so that it is looking for Purchased Date on the correct page of the app.

```
Error: Failed to tap "Purchased Date" TextField: No matches found
for Elements matching predicate '"Purchased Date" IN identifiers'
from input {}

    TextField, placeholderValue: 'Date',
    TextField, placeholderValue: 'Knife',
    TextField, placeholderValue: 'Sharpener'

)}
```

It agreed this was the issue and output new code, which got a bit further, but ran into an actual issue, which was a keyboard was blocking the user's ability to navigate, so I had to add a way to dismiss the keyboard to the screen.

I watched the test execute and it looks like the problem is after the knife name is typed the keyboard is in the way of going to the next step of the test. Please alter the AddKnifeViewController so that there is a way to dismiss the keyboard after typing in a text field.

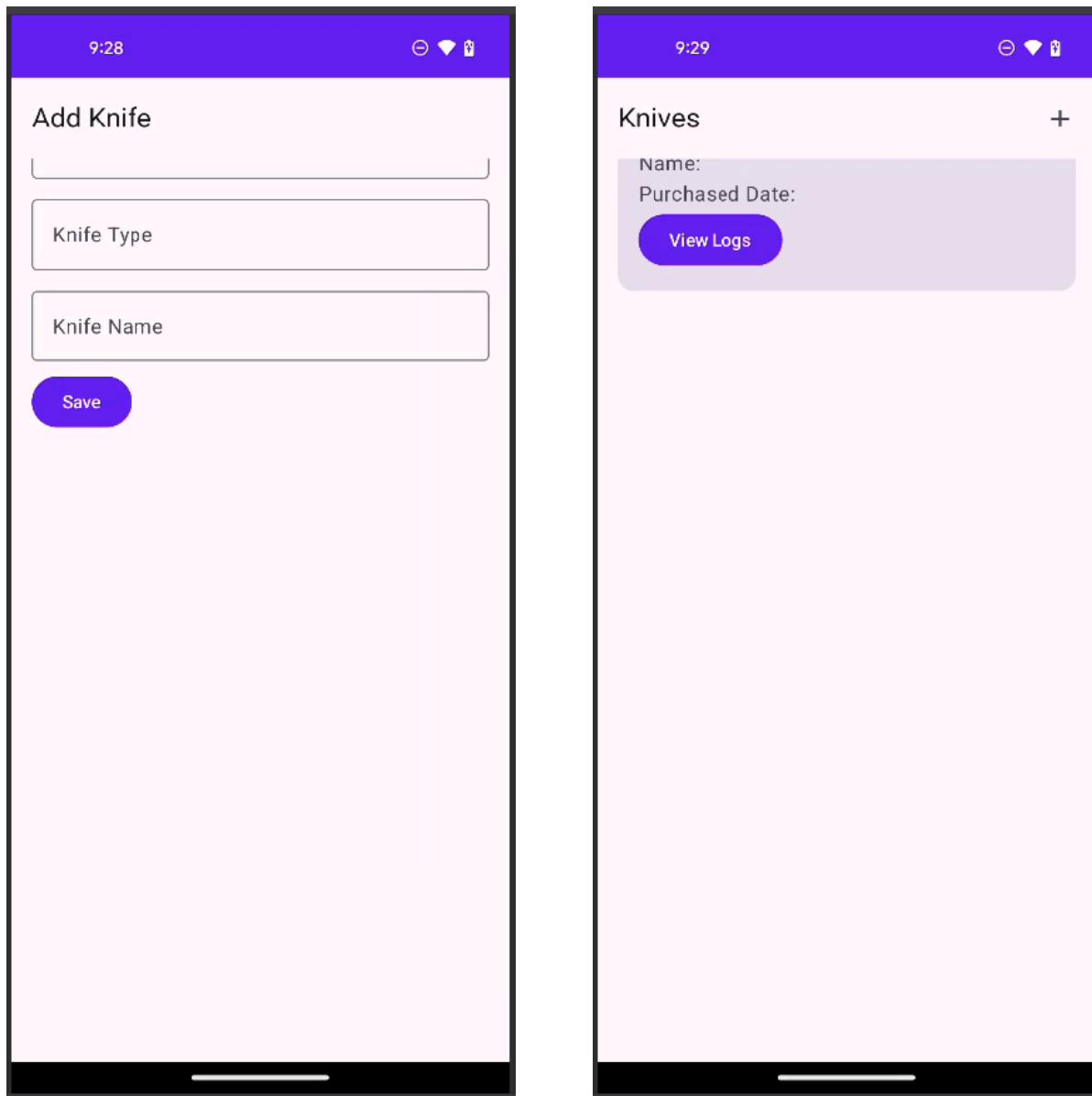
Now the UITests are getting farther, but still failing. I feed in the errors and ask for changes, and run those and feed those in, but after a few cycles of that, it starts making syntax errors and then eventually the test code crashed, so I stopped the experiment. It gets you most of the way there but it's still not able to get working tests for the UI. The data storage ones work fine. You can see the full conversation in the README at the GitHub link above.

Experiment #9: Porting Code

We have a nice iOS app, but what about Android users? Now we will attempt to convert our app into Kotlin using the JetPack Compose framework.

This is great. Can you port it over to a native Android app using Kotlin and Jetpack Compose?

ChatGPT output some code, but no build files. I put the code into Android studio, then asked for the build files. It gave me those and I added them and tried to build. I received an error, which I fed into ChatGPT and copied its improved code over what I had in the repo and tried again. I got more errors and it took about 5 tries to get something that would actually run. It doesn't work, the Knife screen is the only one shown and the View Logs button crashes the app, but I have little doubt I could continue writing prompts until I got what I wanted.



Conclusion

Software Engineers aren't being replaced, they're getting a powerful new tool in their tool belt. The Engineering part: breaking a large problem down into smaller manageable tasks, writing tests, determining the cause of a bug to write better prompts, security and performance considerations, etc. are still needed. The part that is becoming less important is Programming language fluency. I expect interviews to start focusing less on "we need someone to write in X language" and focus "we need someone who can solve these types of problems" and let the AI handle the programming language syntax specifics. It is, after all, a Large **Language Model**.