

Dynamic Input View System

Assignment Spec

Summary

This assignment has two parts:

1. Recreate the UI shown in the attached videos, and
2. Create documentation describing the larger system that this UI would be a part of

Part 1: List Input UI

For the first part, recreate the UI, with a focus on the following core functionalities and UI details:

1. The UI must have both sections, the collection of buttons at the top, and the list of options underneath.
2. The UI must indicate which items are selected and not selected.
3. The UI must allow multiple selection.
4. The UI must allow deselection, both by tapping on a selected row in the list and by tapping on a button at the top.
5. **Note:** The switch doesn't have to do anything. It's just there to demonstrate the dynamic sizing of the table (size to fit content if it's smaller than the available area).

All other visual details (fonts, colors, dots, spacing, sizing, etc) *are not important* – you may style it however you wish. You also do not need to write tests.

We've provided you with a Pronoun model – please use that and the provided `testGroups`. It doesn't matter which `testGroup` you use for your final submission.

Part 2: Architectural Writeup

For the second part, read through the following requirements to understand the larger system that this UI is a part of, and then document how you would design a code system to work with it.

The Ask

In the Hinge app, users need to input many pieces of information to display on their profiles. The Product team wants us to build a flexible, API-configurable system that will allow them to test different input UIs and different input options.

For example, the attached recordings show one way that the app collects users' pronoun information. Based on some configuration object returned by the Hinge API, we must be able to:

- a. *Dynamically generate the UI*
E.g. Instead of a vertical list of options, the Product team may want to test using a matrix of buttons or a text input with search functionality.

b. *Dynamically generate the inputs*

E.g. Instead of using this UI to collect pronouns, the Product team may want to test using it to collect sexual orientation, ethnicity, or religion. The Product team may also add, remove, or change input values.

The Assignment

The assignment is to provide a design doc about the system you would design to accommodate these requirements. What are the building blocks of this system? How do they work together? How will you pattern them together to build different variations of input UI and input options?

You may communicate that documentation however you wish, through words alone or supplemented with drawings, diagrams, flowcharts, etc. This documentation can be limited to the architecture within the iOS codebase (i.e. you don't need to document what the configuration object looks like or how you retrieve it, unless you feel it helps you communicate your vision).

Additional Notes

- You don't have to code Part 1 of this assignment in a way that conforms to the architecture you outline in Part 2. However, using it as a reference point in your discussions might help illustrate your vision more clearly.
- Please timebox this assignment to around 5 hours max. It doesn't have to be perfect (especially Part 1 – we're more interested in how you communicate your system design in Part 2).
- You may use any technologies you want, e.g. SwiftUI, Combine, etc, but keep in mind that you will be asked to build on top of your project if you proceed to the next interview stage. Please set yourself up for success.
- We've provided you with a starter project. You don't have to use it, but note that it contains the Pronoun struct that you may want to use for reference.