**Context**

Given the variety of users we support and come across, it’s important to ask the user questions about their finances to determine the right kind of financial advice.

We try to ask questions when appropriate (i.e. when we can demonstrate value off the information asked vs just being creepy), but that can still involve several upfront questions. As one can imagine the questions we ask tend to be followed up with further contextual questions. For example, if a user is married or single that can in theory have a drastic impact on the remaining questions we may ask during, for example, a registration/initial onboarding flow.

What we’d like to build is a system around asking questions and having some semblance of a dynamic flow based on questions answered.

**Technical Requirements Description**

Here’s a sample of example question types we ask (you can assume this is all you need to support for this exercise) :

* Textual
  + “What is your first name?”
  + “What is your job title?”
* Numerical
  + “How many kids do you have?”
  + “What is your monthly salary?”
* Pre-determined Options (textual or numerical)
  + Multi-select options
    - “What do you worry the most about?”
      * Saving for Retirement, Paying Off Debt, Buying a Home, Getting Married, Budgeting Better, Investing, Earning More
  + Single-select options
    - “Are You Married?”
      * Yes, No
    - “What is your debt range?”
      * $0-$500, $500-$2500, $2500-$10000, Greater than $10000

On the more visual side, the design team is already thinking through that they may want to ask several questions on one screen (it may or may not need to scroll then), while for other questions they might want to ask one question at a time (for example, they might come up with a cool animated/special way to ask salary if it’s taking up the whole screen).

**As a sample flow to think through, Flow #1, the product marketing design team could envision wanting to explore:**

* **Screen #1 - 3 questions asking: name, job title, geographic location**
* **Screen #2 - 1 question asking monthly salary**
* **Screen #3 - 1 question asking about your biggest financial worries**
* **Screen #4 - questions about debt, monthly fixed costs (bills), etc based what Screen #3 was selected**

**Exercise Task**

Given the nature of mobile apps: 3rd party review process, being permanently installed to a device and wanting a system that can be configured it’s in our advantage to build something we can control from our end server-side. That way design, product and engineering can make changes to it in the future without a re-release, opportunities to A/B test, etc.

Determine a system to provide an app with a way to describe questions experience that is derived from the requirements above. You are free to use Flow #1 as a model to actually implement or do something else that demonstrates its requirements, feel free to be creative! The sample project should:

1. **Basic**
   1. Provide an implemented strategy for receiving content from the network (i.e. you don’t have to build the server app, but rather you should mock out a response that your app will receive).
   2. Interpret this server-side content in a way that can produce a flow based on these idea of screens with questions
   3. Implement a first pass of getting these screens to connect to each other (the final screen can land you on a empty view)
2. **Intermediate**
   1. Allow for questions’ answers to be referenced by other questions, for example:
      1. In Flow #1 once we know the name of the user we might want the experience to say something like “Hi John! Let’s talk about…” at the top of subsequent question screens.
      2. Add a way for our system to have “takeaway” sections, i.e. based on set of questions’ answers have a way for the user to see some benefit/analysis from their answer. For example, if the user says they’re 25 and have more than $100k in savings (2 questions a flow could’ve asked) we could say something like “Great job! Did you know most people in their 20’s only on average have $1k in their bank account” or if they’re 25 and have $750 saved “It’s ok! A lot of people in their 20’s are exactly where you are. This is why we’re here, let’s work on saving together.”
3. **Advanced**
   1. Have a way for individual questions or entire screens to be bypassed/skipped (this would likely involve 2a being done first).
      1. For example, in Flow #1 on Screen #3 if the user has chosen that debt is a big financial worry, we’ll ask for debt in Screen #4 otherwise we won’t ask for it. Likewise if they choose something else
   2. Provide a way for the questions and their screen to animate into each other with something other than a UINavigationController push animation for (Be creative!). For example, the designers on this project were imagining that the whole screen doesn’t need to change so that something like that “Hi John! Let’s talk about…” treatment would stay fixed at the top.

**What We’re Looking For**

* Does it perform as expected?
* How flexible is the architecture?
* Is the code readable? maintainable? testable?

Good luck and have fun with it!