Assignment

Build a simple Survey app. It will consist of **two screens**. Example wireframes are provided at the end of the document.

Initial screen

The **initial screen** will have a button “Start survey”. Pressing the button will load the second screen that will display the list of survey questions.

Questions screen

The second screen will be a horizontal pager of all the questions.

- Pressing **previous and next buttons** moves to the previous and next question. Previous button should be disabled when on the 1st question and the next button should be disabled when on the last question.
- Pressing the **submit button** posts the request to the server. Submit button is disabled when no answer text exists or when the answer has been already answered.
- A **counter** of already submitted questions exists on top of every question. It should be updated dynamically after every successful question submission.

Note: Tapping back (going to the Initial screen) and then tapping “Start survey” should restart the survey with 0 answered questions.

Submit a question

Submit question endpoint can:

- (1) Succeed with a 200 status response
- (2) Fail with a 400 status response

Upon **success** (1), a **notification banner** should appear for some seconds with text: “Success!”
Upon **failure** (2), a **notification banner** should appear for some seconds with text: “Failure….” and a retry button!

Successful submission of questions should be kept in memory so that when navigating to an already submitted question, the submitted answer will be present and the submit button will be disabled.

Note: This does not imply any offline storage. It “only” means that the question submission status is kept in memory while you navigate back and forth amongst the questions.

What we would like to see

Do not waste your time on fancy UI. We are more interested in your code structure decisions. And since our team heavily invests on multiple methods of software verification, unit tests or any other form of testing would be highly appreciated.

[iOS candidate] What we would also like to see

The task should be implemented using:

- FRP (e.g. Combine, ReactiveSwift, RxSwift) **or**
- Swift Concurrency **or**
- a unidirectional data flow style (e.g. TCA)

The reason is that we use these in our codebase and we want to see that the candidate is comfortable in at least one of these areas.

UI should ideally be implemented in SwiftUI, since all new UI in the app is written in SwiftUI.

[Android candidate] What we would also like to see

The task should be implemented using:

- FRP (e.g. RxJava, Flows) **or** Kotlin Coroutines
- In a unidirectional data flow style (e.g. MVI or TKA).

The reason is that we use these in our codebase and we want to see that the candidate is comfortable in at least one of these areas.

UI should ideally be implemented in Compose, since all new UI in the app is written in Compose.

How to share the exercise with us

We prefer the Github repo approach, since this makes it explicit to the candidate to create a git repo. This allows you to showcase your commit style, add a README if you want etc (all these are optional)
If you want to keep the repo private, you can add our team members:

- for iOS use nlinakis-xm / aloukas-xm
- for Android use aloukas-xm / dkonomi-xm

FAQ

Q: Is it ok to use a 3rd party library?
A: Yes, it's ok :)

Q: Do I need to persist anything offline between app launches?
A: No, nothing needs to be persisted offline.

Q: Do we have to use any specific architecture?
A: No, as long as we see what we would like to see.
BUT, since you ask, using TCA on iOS / TKA on Android will be considered a plus!

Endpoints

Base URL: https://xm-assignment.web.app

GET /questions
Example response:

[
{"id": 1, "question": "What is your favourite colour?"},
{"id": 2, "question": "What is your favourite food?"}
]

POST /question/submit

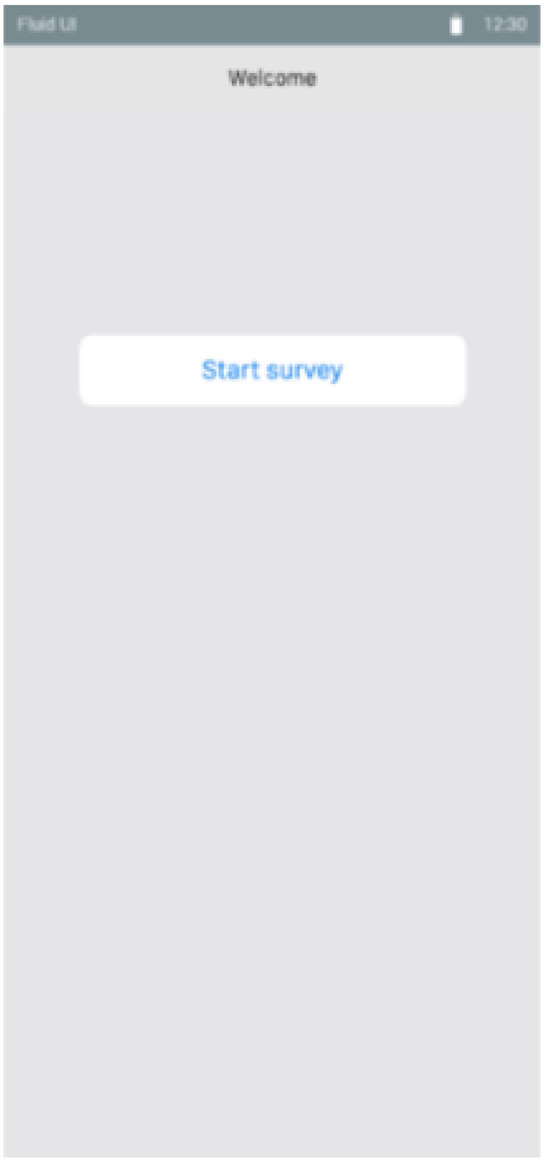
{
"id": 1,
"answer": "red"
}

Example response:
Empty response with either 200 or 400 HTTP status code

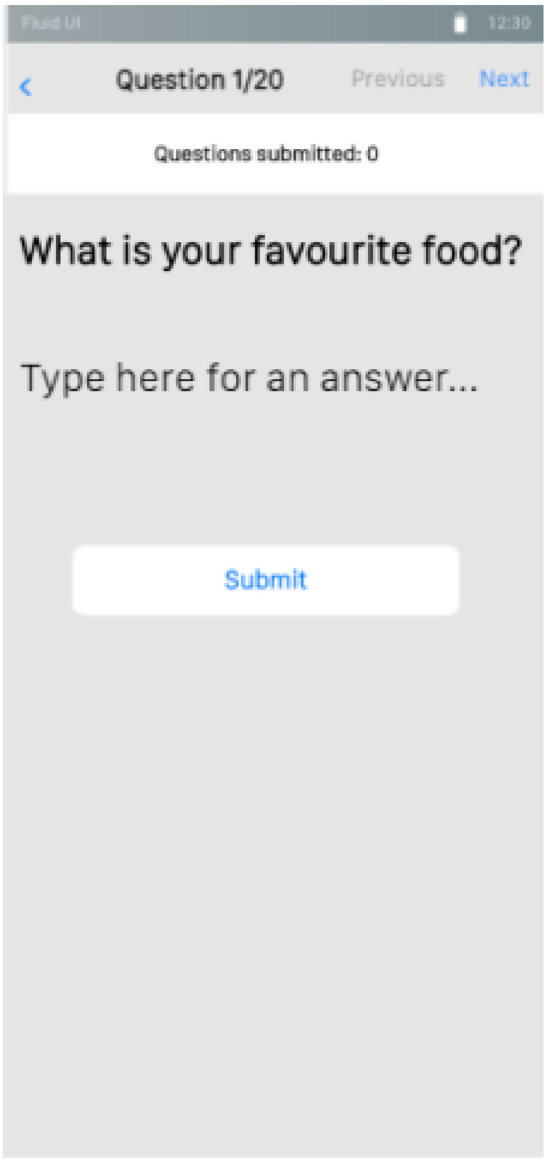
REMARK:
Because of Firebase's Functions cold boot the very first request may fail until the instance is awakened.

Wireframes

Initial screen

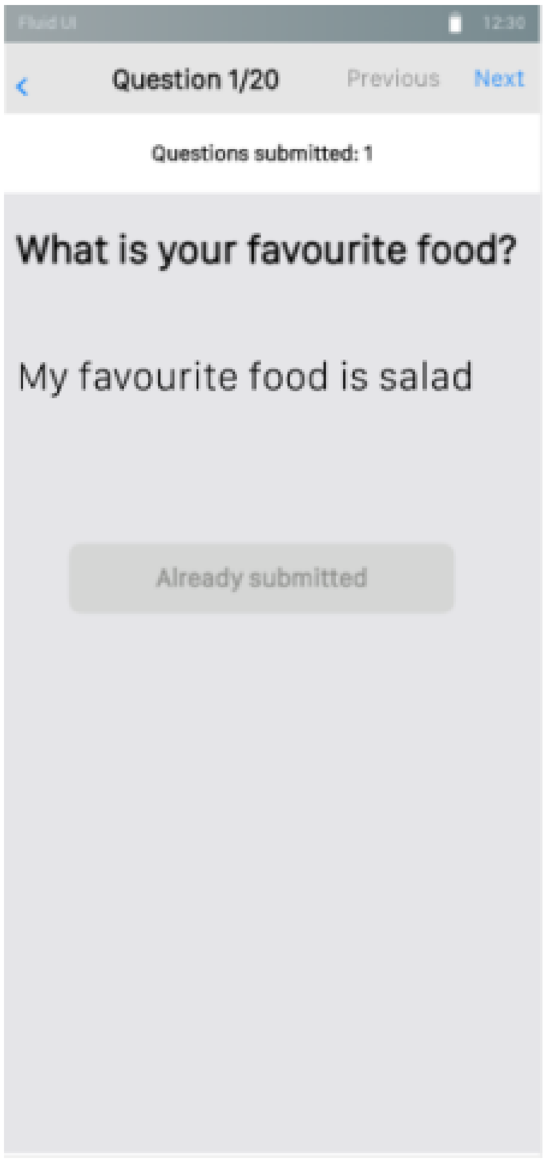


Questions screen



What is your favourite food?

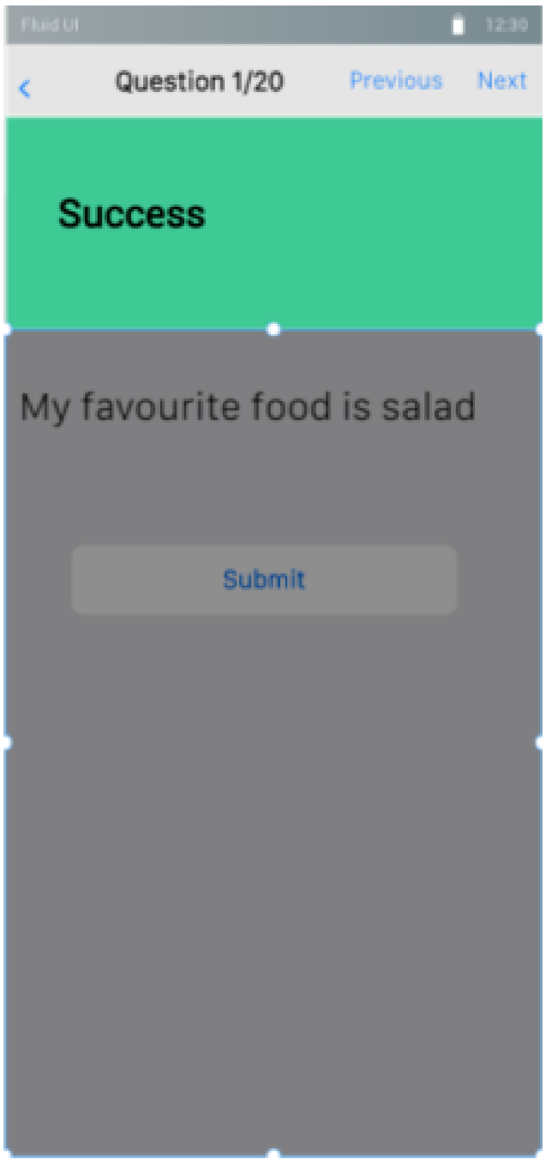
Questions screen
(already submitted question)



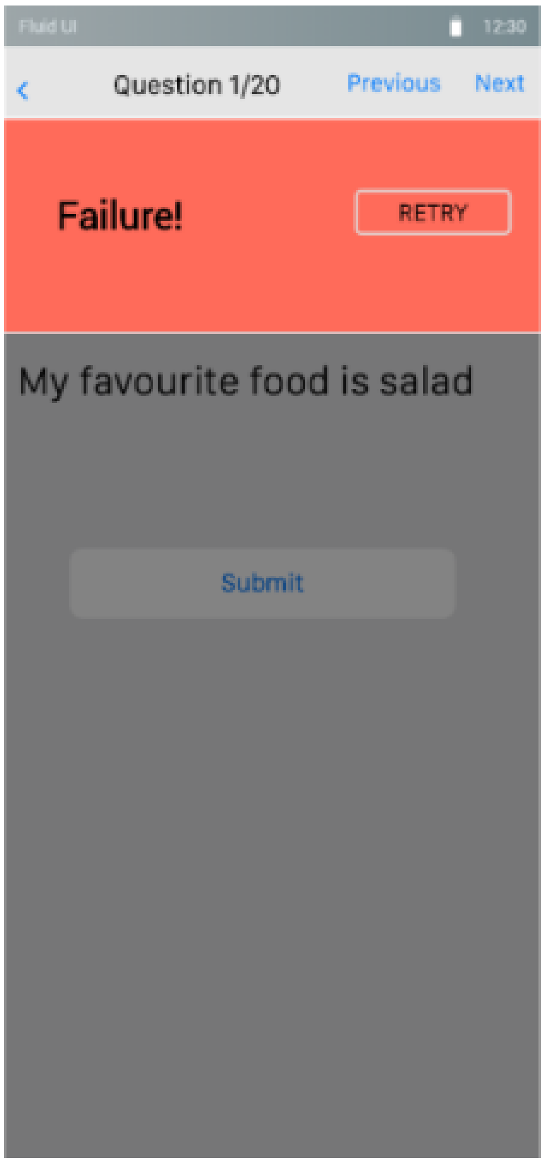
What is your favourite food?

My favourite food is salad

Successful submission



Failed submission


Fluid UI
12:30
Question 1/20
Previous
Next
Failure!
RETRY
My favourite food is salad
Submit