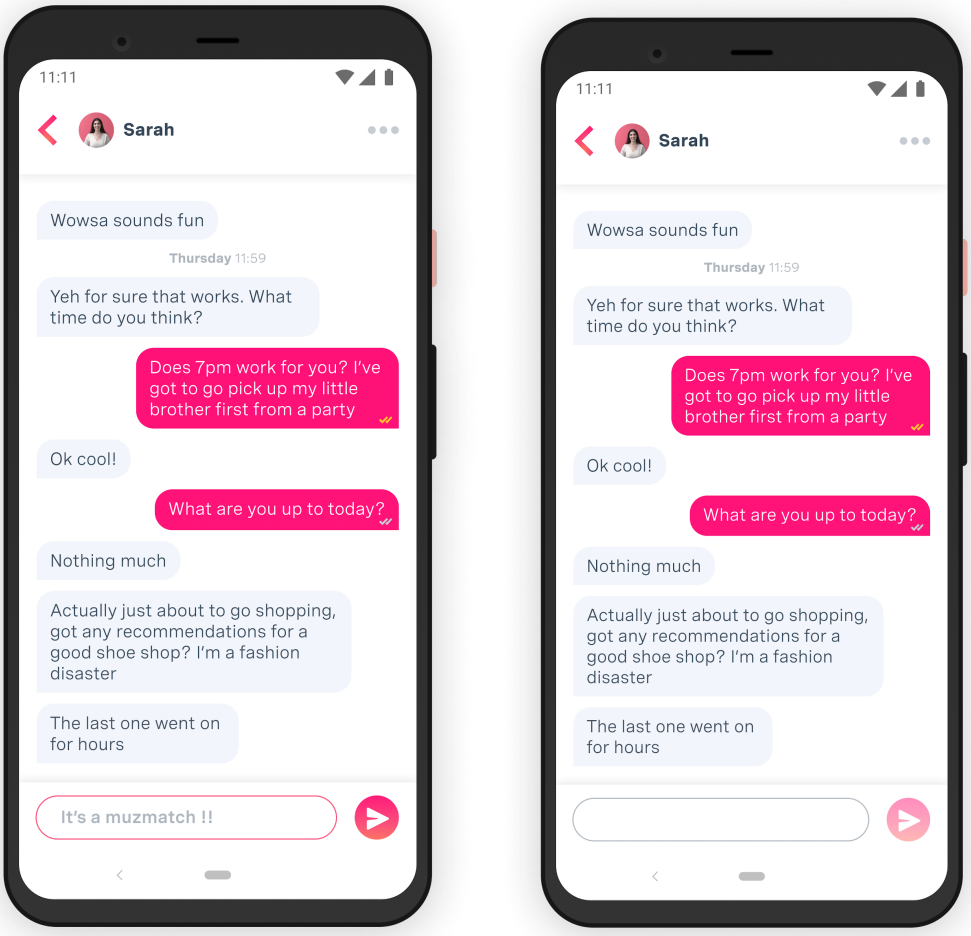
Android Exercise

Context

This exercise is deliberately open ended, you can spend as little or as long on the exercise as you want. There is no obligation to spend more than an evening on this exercise but we encourage you to take as much time as you feel you need.

We are concerned with code quality, brevity and logic used as well as the quality of your UI implementation.

This should not take an undue length of time to create.



Challenge

In this exercise you are expected to create a chat interface similar to that found in the Muzz app.

Message List and Text Entry Box

Create a Message List with a Text Entry box as per Screenshot 1. This Message List's contents should display the newest items at the bottom and the oldest at the top.

As you enter text into the Text Entry Box and tap "send" the item should be added to the Message List.

Message List items

Distinguish between received and sent messages by aligning them left or right.

The messages should be encased in a "bubble" as per Screenshot 1.

A message has a tail when any of the following 3 criteria are met:
It is the most recent message in the conversation
The message after it is sent by the other user
The message after it was sent more than 20 seconds afterwards

Item sectioning

If a previous message was sent more than an hour ago, or there is no previous messages, it should be sectioned as in the screenshots (Thursday 11:59). The format for this is "{day} {timestamp}".

Bonus Points: Persistent storage and observables

An additional challenge we'd like to see Mid-Level and Senior delivering is to implement persistent storage of messages in a database, and an architecture allowing UI classes to observe data form the database so that it always displays the latest information.

Two way messages

The app should have some way of being able to trigger messages from the “other” user. How you do this is up to you, you can use random replies, a toggle to switch users, or whatever else you think might work.

Not required

There is no need to have any external network connectivity to get messages from an external source. All messages in the chat should be created (or pre populated) locally on the device.

There is no need to provide any form of login activity to support multiple users in the app. The app should be a single activity between two fixed users.

Deliverables

Please deliver a ZIP with the following:

- Your Android Project
- PDF outlining implementation decisions, app limitations and what you would do given more time
- Short recording of the app running on an emulator or on the device
- Ideally: Github repo link showing your commits

What are we looking for?

Be sure to:

- Detail all assumptions made in your implementation and decisions in your readme
- Show good architecture, brevity of code, comments and understanding of the problem
- Show you are a solid mid/senior engineer
- Show off some good language techniques you are aware of - always good to stand out in a good way!
- Don’t over engineer things - KISS.

What will I get back?

We pride ourselves on respect and speed of decision making to you as a candidate. We promise to:

- Have senior engineers review your exercise ideally within 24-48hours
- Provide detailed USEFUL feedback of the good, the bad and the ugly of your solution
- Progress to final interviews if we deem the exercise to be satisfactory
- Decision 24 hours after final interviews