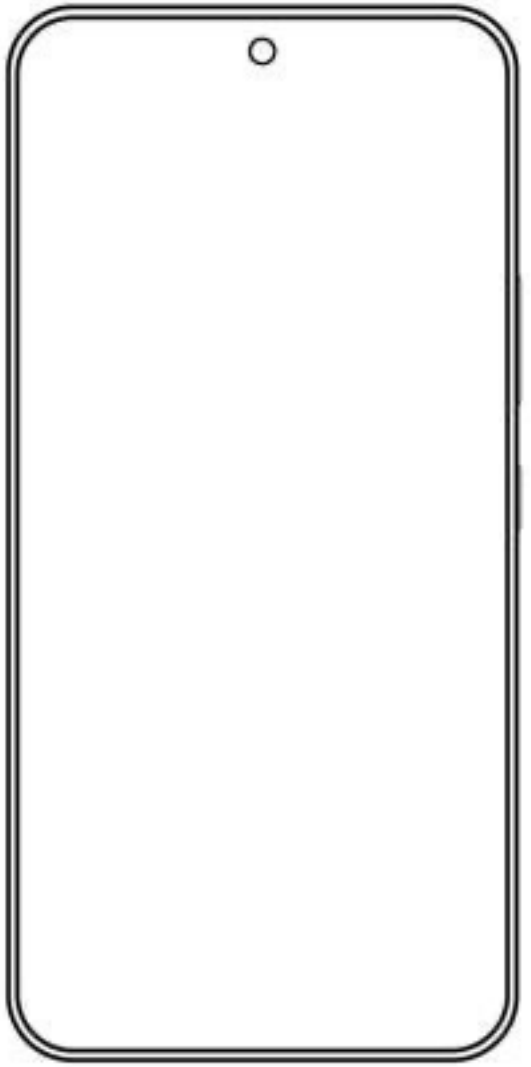
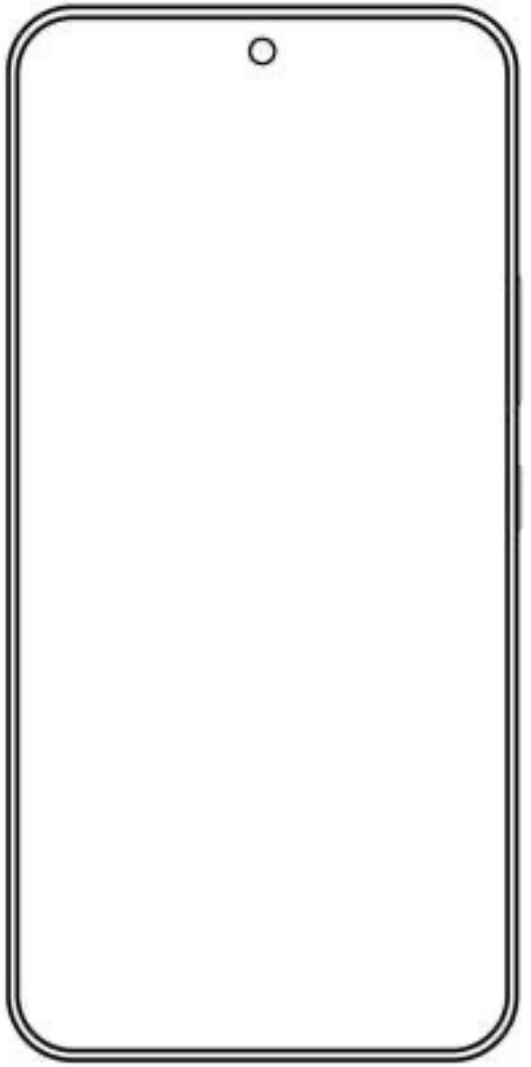
Android USB Chat

A functional application in Kotlin to understand Android native USB.



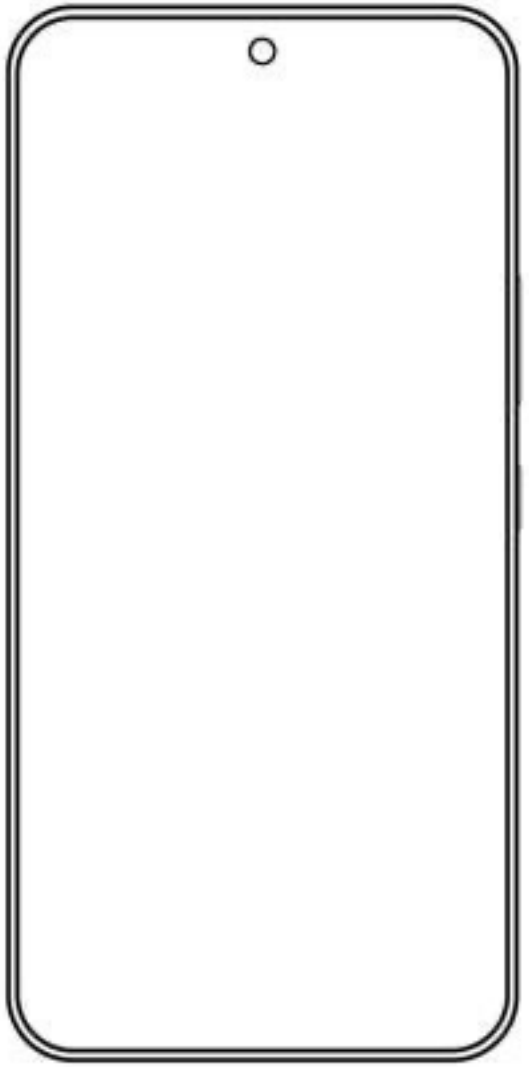
Purpose

- Develop a functional application in Kotlin to understand Android native USB basics
- Solution is comprised of two Android devices connected using an OTG USB-C to USB-C cable
- The application enables the two devices to have a simple chat session, allowing messages to sent back and forth between the two devices over the USB connection



USB Basics

- Even though the two devices will have the exact same functionality, as required, one may be considered the Server and the other the Client.
- The application should support several features to highlight USB features outside of the basic chat:
  - USB device filtering by PID/VID
  - USB device connect / disconnect event detection
  - USB device current status detection (connected / disconnected)
  - USB device error conditions



Basic UI Concepts

- App Title
  - The application name
- Connection Status Label
  - Shows “Connected” or “Disconnected”
- Chat History Text Area
  - A list of all the messages, with scroll bar. Last message received will be shown at bottom, like text message app
- Send Text Area
  - The area where the user types in a new message to be sent
- Send Button
  - The button which will activate the send function
- Check Status Button
  - Gets the connection status, real-time check, and prints in History

Basic Function Concepts

- When device is not connected, the Connection Status Label will show “Disconnected”
- When device is connected, the Connection Status Label will show “Connected”
- If Disconnected, all text areas and button will be disabled.
- When a message is received, the message will be displayed at the bottom of the Chat History Text Area
- Each subsequent message will be displayed at the bottom of the Chat History Text Area
- To send a message, the user will type the message in the Send Text Area and then hit the Send Button.
- To perform a real-time connection status check, the user will press the Check Status button and a message with the status will be inserted at bottom of Chat History Text Area.

Code Concepts

- Kotlin, Gradle, Minimum SDK 26, Target SDK 33
- All events should be logged to the logger
  - Connect
  - Disconnect
  - Message Received, Bytes Received
  - Message Sent, Bytes Sent
  - Error events
- Message history does not have to be saved
- Comments in the code are requested in areas of USB functionality