

# Technical Requirements – Still Connected

Challenge is accessible on Memberspot: <https://pl-coding.mymemberspot.io/library/jx3b7Qik9ip5qpNI8IF2/JvkX6luGnKXnd8tkr5WR/IO0707yenqrrsCmG3Wzl/details>

## Scenario

This mini-challenge focuses on building a simple single-screen application that reflects the device's current internet connection state in real time and clearly communicates whether the user is connected, has lost the connection, or has intentionally disconnected by enabling Airplane Mode.

## Figma Mockups

<https://www.figma.com/design/V89HDDOUngkPG558FrZoNm/February-Moments?node-id=2-214>

## Font - [Stack Sans Headline](#)

## Feature Goal

The goal of this challenge is to implement a reactive UI that accurately reflects the device's connectivity state using Android system APIs and updates automatically when that state changes.

## Requirements

### Screen Structure

- The app consists of a **single screen**.
- The app can be in one of three states: **Connected, Connection Lost, Disconnected by you (Airplane Mode)**.
- The screen has the **same structure** across all states.
- Each state differs only by:
  - a central visual element
  - a textual status message
- Visual representations of all states can be found in the Figma mockups.

## State Priority Rule

- If Airplane Mode is enabled, the **Disconnected by you** state must always be shown, regardless of internet availability.
- If Airplane Mode is disabled, the state is determined by whether an internet connection is available.

## Connected

- Displayed when:
  - Airplane Mode is disabled
  - internet connection is available
- Status text: *"You're connected"*



You're Connected

## Connection Lost

- Displayed when:
  - Airplane Mode is disabled
  - internet connection is not available
- Status text: *"We lost the connection"*



We Lost the Connection

## Disconnected by You (Airplane Mode)

- Displayed when:
  - Airplane Mode is enabled
- Status text: *"You turned on airplane mode"*



You Turned on Airplane Mode

## Technical Behavior

- The connection state must be determined using Android system APIs.
- The app must react to:
  - changes in internet connectivity
  - enabling or disabling Airplane Mode
- State observation must be lifecycle-aware.

## 🤔 What's Allowed?

- Standard Android/Jetpack libraries
- No 3rd party libraries are allowed or would be required to complete this challenge
- Using standard Android connectivity APIs.

## What's not important

- Responsiveness across every device size or orientation is not mandatory.
- Light / Dark mode.
- Complex animations.
- Handling every possible network edge case.

## Useful Links for This Challenge

- [Monitor connectivity status and connection metering](#)
- [How to Observe the REAL Internet Connectivity](#)
- [Stateful vs. Stateless Composables](#)
- [State Hoisting in Compose](#)
- [Managing State in Jetpack Compose \(Codelab\)](#)

## Submission & Rewards

- Successfully submitting this challenge via the `/submit-challenge` command on Discord grants you **100 XP**.
- Your submission must include:
  - a. A **Gist link** with your implementation.
  - b. A **screen recording** (max 20 seconds) showing:
    - The Connected state.
    - Disabling internet → Connection Lost.
    - Enabling Airplane Mode → Disconnected by you.
    - Automatic UI updates without restarting the app.