

Technical Requirements – Last Active

Challenge is accessible on Memberspot: <https://pl-coding.mymemberspot.io/library/jx3b7Qik9ip5qpNI8IF2/JvkX6luGnKXnd8tkr5WR/6MBd6BfvdcJrFw6SDkfW/details>

Scenario

This mini-challenge focuses on building a simple single-screen interface that displays a user's activity status in a "Last active" format, showing the moment when the app last transitioned to the background. For simplicity, the app displays the user's own activity status, while the logic mirrors how another user's presence is typically shown in real messaging and social apps.

Figma Mockups

<https://www.figma.com/design/V89HDDOUngkPG558FrZoNm/February-Moments?node-id=0-1>

Font - [Figtree](#)

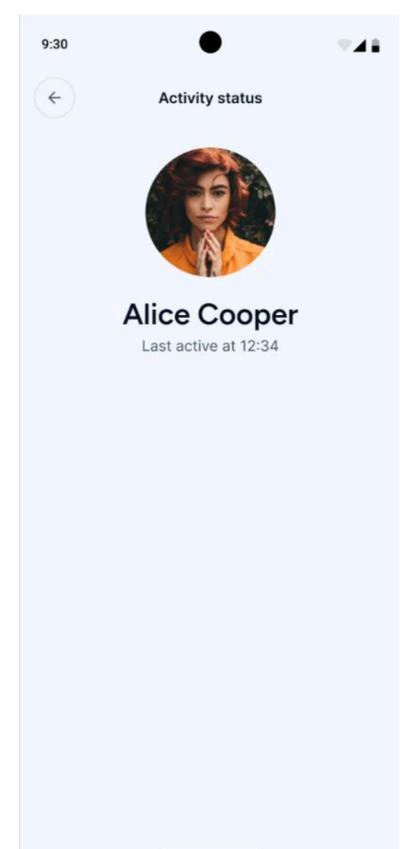
Feature Goal

The goal of this challenge is to correctly detect when the app moves to the background, store the timestamp of that moment, and restore and display the saved activity status on every app launch.

Requirements

Top Bar

- Positioned at the top of the screen.
- Contains:
 - a Back button
 - the screen title "Activity status"
- The Back button is decorative only and has no functional behavior.



Profile / Context Block

The block includes:

- the user avatar (as provided in the Figma mockups)
- the user name: Alice Cooper
- a single-line activity status displayed below the user name

Layout, spacing, and visual styling must match the provided Figma mockups.

Activity Status

- The activity status is displayed as a single line of text under the user name.
- The screen supports two possible states:

No Activity Yet

- Displayed:
 - on the first app launch
 - when no background transition has been recorded yet
- Status text: *"No activity yet"*

Last Active

- Displayed when:
 - the app has transitioned to the background at least once
- The activity status text adapts based on how recent the last activity was:
 - If the activity occurred today: *"Last active at HH:mm"*
 - If the activity occurred yesterday: *"Last active yesterday"*
 - If the activity occurred earlier: *"Last active on MMM dd"*
- The timestamp:
 - is recorded when the app moves to the background
 - does not update when the app is opened again
 - is displayed in a clear and readable time format

State Persistence & Restoration

- The last activity timestamp must:
 - persist across app launches
 - be restored every time the screen is opened
- After restarting the app:
 - the previously saved activity status must be displayed correctly
- If no activity has been recorded:
 - the *"No activity yet"* placeholder state is shown

🤔 What's Allowed?

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

⚠️ What's not important

- Responsiveness across every device size or orientation is not mandatory.
- Light / Dark mode.
- Complex date localization or advanced time formatting.
- Handling edge cases such as time zone changes.

🔗 Useful Links for This Challenge

- [Lifecycle of composables](#)
- [ProcessLifecycleOwner](#)
- [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:
 - Opening the app for the first time and displaying the “*No activity yet*” state.
 - Sending the app to the background (e.g. pressing the Home button).
 - Reopening the app and showing the “*Last active ...*” status.