Technical Requirements – Moment Journey Flow

Challenge is accessible on Memberspot: https://pl-coding.mymemberspot.io/library/jx3b7Qik9ip5qpNI8IF2/JvkX6luGnKXnd8tkr5WR/cCDFrMxTUg601QsBTjK0/details

🎭 Scenario

This challenge focuses on building a Moment Journey app that lets users capture and revisit personal moments in a chronological timeline. The app can be launched from multiple entry points—such as the app icon, a home screen widget, notifications, or external intents—and each entry point must lead to a specific screen with a correctly constructed navigation flow and back stack.

🎨 Figma Mockups

https://www.figma.com/design/V89HDDOUngkPG558FrZoNm/February-Moments?node-id=13-968

🔤 Fonts - Stack Sans Headline, Inter

🎯 Feature Goal

The goal of this challenge is to implement explicit, entry-point–aware navigation, ensuring that the back stack is built correctly and the Back button behaves predictably based on how the app was launched.

📌 Requirements

Journey Overview Screen

Journey Overview is the main screen of the app that displays a chronological list of the user's saved moments.
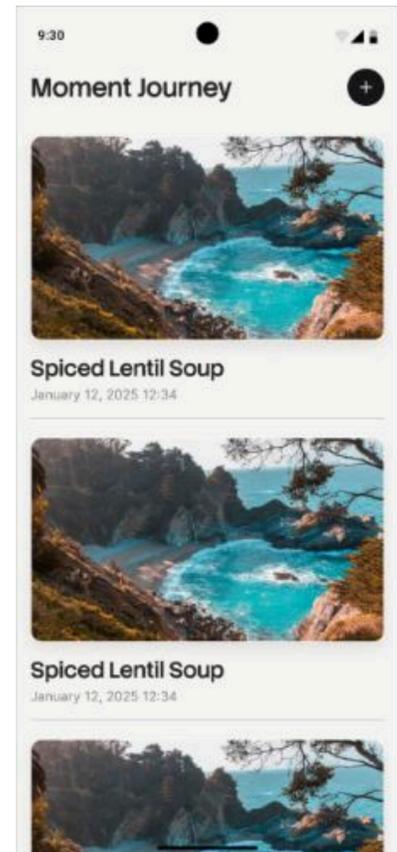
Top Bar

- Positioned at the top of the screen.
- Contains:
  - the app title *"Moment Journey"*

- an **Add (+)** button on the right
- Tapping the Add button opens the **Add Moment** screen.

Content

- The screen displays a **vertical list** of moments.
- Each moment in the list includes:
- a preview image
- a title
- the creation date and time of the moment
- The date and time display when the moment was created and must be shown in the following format: January 12, 2025 12:34.
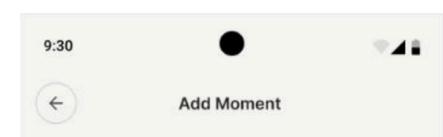- Tapping a moment in the list opens the corresponding Moment Details screen.

Add Moment Screen

Add Moment is a screen for creating or editing a moment, where the user can add a photo, enter a title and description, and save their changes.

Top Bar

- Positioned at the top of the screen.
- Contains:
  - a Back button
  - the title *"Add Moment"*
- The Back button returns the user to the previous screen.

Content

The screen contains the following elements:

- a photo area for the moment
- an **Add Photo button**. Tapping this button opens the system image picker, allowing the user to select a photo from the device.
- a title input field
- a description input field
- a primary action button at the bottom (Save or Edit, depending on the state)

The content of the screen is vertically scrollable so that input fields remain accessible when the keyboard is visible. The screen has three main states:

Empty State

Shown when the user is creating a new moment and has not entered any data yet.

- The photo area displays a placeholder with the hint *"Add Photo to start your memory"*
- Input fields:

- Add Moment Title
  - Add Description
- The Save button is **disabled**.

Typing State

Shown when the user is entering text.

- The keyboard opens automatically when an input field gains focus and does not shift the Save button upward.

- Keyboard actions:

  - In the Title field: the keyboard shows a **Next action** that moves focus to Description.

  - In the Description field: the keyboard shows a **Done action** that closes the keyboard.

- While typing:

  - the content scrolls as needed to keep the input fields visible

  - the Save button may be covered by the keyboard and is **not expected to remain visible**

Save Behavior

- A new moment is created

- The **creation date and time** are recorded

- After saving:

  - the user is returned to **Journey Overview**

  - the moment appears in the list

  - a **standard local notification** is shown
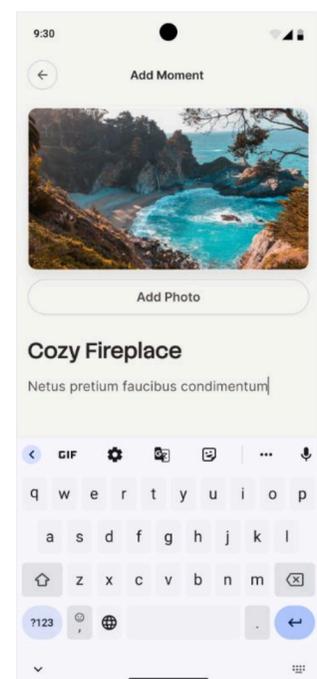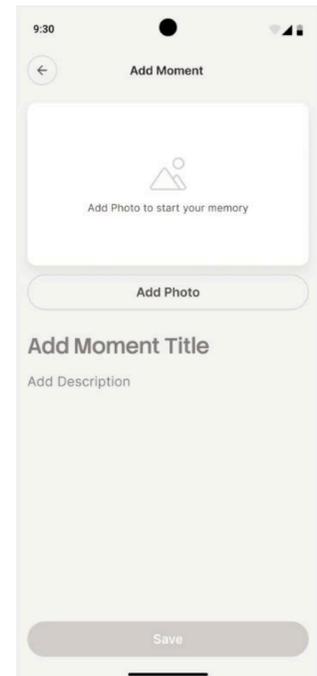
Notification Details

- Title: *Moment saved*

- Message: *Tap to view your moment*

- Tapping the notification opens Moment Details for the saved moment.

Moment Details Screen

Moment Details is a screen that shows detailed information about a single moment.

Top Bar

- Positioned at the top of the screen.

- Contains:

  - a Back button

  - the title *"Moment"*

- The Back button navigates to the previous screen according to the defined navigation flow.

Content

- The screen is vertically scrollable.

- The main image is displayed at the top of the screen and has rounded corners.

- The title is shown directly below the image.

- The description displays the full text of the moment.

- The **date and time** show when the moment was created, using the same format as on the Journey Overview screen.

- A primary action button labeled **Edit** is displayed at the bottom of the screen.

- Tapping Edit switches the screen into **Edit State**.

Edit State

Edit State is shown after the user taps Edit on the Moment Details screen.

- The photo, title, and description become editable.

- The primary action button label changes from Edit to **Save.**

Save Behavior

- When Save is tapped:

  - the moment is updated with the new values

  - the original creation date and time remain unchanged

  - no notification is shown

  - the screen returns to the default (read-only) Moment Details state with updated content displayed

Import Moment — Behavior & UI Specification

This modal appears when the app receives an external image (e.g., via the Android Share Sheet, "Share" from the gallery, or "Open with"). It is shown immediately after the app receives an implicit intent containing an image.
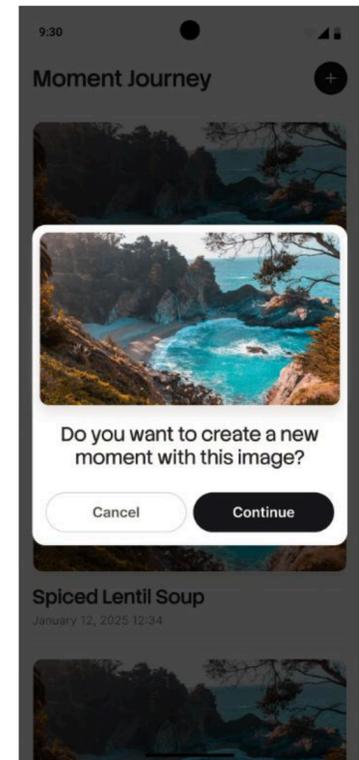
Modal Behavior

- The modal overlays the current screen with a dimmed background.

- Tapping outside the modal does not dismiss it.

- Pressing the system Back button dismisses the modal.

Content

- Displays the imported image with rounded corners.

- Confirmation text: *"Do you wanttocreate anew momentwith this image?"*

- Actions:
  - Cancel
    - Closes the modal and returns to the previous screen.
  - Continue
    - Opens the Add Moment screen.
    - Pre-fills:
      - the photo (already selected)
      - Title — empty
      - Description — empty
    - The rest of the flow behaves exactly like manual moment creation, **except that the image is already selected.**
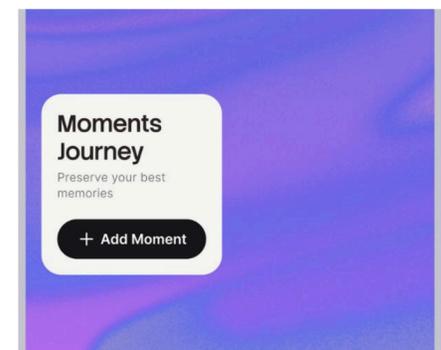
Error Handling

Handling import errors is left to your discretion. If desired, a minimal fallback may be a short message such as: *"Unable to import image"*

Home Widget

The widget displays the following content:
- Displays:
  - Title: *Moments Journey*
  - Subtitle: *Preserve your best memories*
  - Primary Button: *Add Moment* with a leading "+" icon
- Tapping the widget opens the **Add Moment** screen.

🧠 App Behavior Logic

The app must explicitly control its navigation stack depending on the entry point, without relying on the system's default behavior.

Standard Launch (App Launcher)

- Opens **Journey Overview**.
- Back → exits the app.

Launch from Home Widget

- Tapping the widget opens **Add Moment**.
- Back from Add Moment → **Journey Overview**.
- Back from Journey Overview → exit.

External Import (Share Intent)

- When receiving an image from another app, **Import Moment** is shown — a one-time confirmation step that is not part of the navigation stack.

- **Cancel / Back** on Import Moment → exits the app.

- **Continue** → opens **Add Moment** with the imported image pre-filled.

- Back on Add Moment → **Journey Overview**.

- Back on Journey Overview → exit.

Launch from Local Notification

- Tapping the notification opens **Moment Details** for the corresponding moment.

- Back from Moment Details → **Journey Overview**.

- Back from Journey Overview → exit.

🤔 What's Allowed?

- Standard Android/Jetpack libraries
- No 3rd party libraries are allowed or would be required to complete this challenge
- Using WorkManager, NotificationManager, or any other standard Android APIs to trigger the local notification.

⚠️ What's not important

- Persistence or long-term storage of moments (in-memory storage is acceptable for this challenge).
- Custom notification layouts — a standard system notification is fully sufficient.
- Handling edge-case errors for Share Intents
- Animations, transitions, or custom gesture behavior.
- Complex widget behavior — the widget only needs to open the Add Moment screen.
- Responsiveness across every device size or orientation is not mandatory.
- Light / Dark mode.

🔗 Useful Links for This Challenge

- [Navigation with Compose](#)
- [Photo picker](#)
- [App widgets overview](#)
- [Start an Activity from a Notification](#)
- [Intents & Intent Filter video](#)

- [Intents and intent filters documentation](#)
- [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 **300 XP**.
- Your submission must include:

  a. A **Gist link** with your implementation.

  b. A **screen recording** (max 40 seconds) showing:

  i. External Import
  - Share an image to the app from another app (e.g. Gallery).
  - Show the Import Moment modal.
  - Tap Continue to open Add Moment with the image pre-filled.
  - Press Back to return to Journey Overview.

  ii. Home Screen Widget
  - Tap the Add Moment home screen widget.
  - Verify that Add Moment opens.
  - Press Back to return to Journey Overview.

  iii. In-app Creation
  - From Journey Overview, tap the Add (+) button in the top bar.
  - Create a new moment and tap Save.
  - Verify that:
    - you return to Journey Overview
    - a local notification is shown

  iv. Notification Navigation
  - Tap the notification.
  - Verify that Moment Details for the saved moment opens.
  - Press Back → return to Journey Overview.
  - Press Back again → the app exits.