Technical Requirements – Shared Valentine

Challenge is accessible on Memberspot: https://pl-coding.mymemberspot.io/library/jx3b7Qik9ip5qpNI8IF2/JvkX6luGnKXnd8tkr5WR/peV8ZSuSrrkLd237yrLZ/details

🎭 Scenario

This challenge focuses on building two Android apps that interact through the standard system file sharing flow. One app allows the user to select a Valentine card and send it, while the other app receives the shared image, displays it in a preview, and lets the user save or discard it.

🎨 Figma Mockups

https://www.figma.com/design/V89HDDOUngkPG558FrZoNm/February-Moments?node-id=7-70

🔤 Font - Stack Sans Headline

🎯 Feature Goal

The goal of this challenge is to implement a clear and reliable send–receive flow between two apps using standard Android file sharing mechanisms, while correctly handling received content and updating the UI based on the current app state.

📌 Requirements

Project Structure Note

For this challenge, both apps may be implemented within a **single Android project** using Configure product flavors.

Each flavor (Sender and Receiver) behaves as a separate app:

- it is installed independently on the device
- it has its own app name, launcher icon, and entry screen
- it implements its own UI flow

At the same time, you may share common visual styling and resources across both apps, such as fonts, colors, and themes. The screens and business logic for Sender and Receiver, however, should be implemented as two independent flows within the same project. This approach allows you to build and test two full-featured apps without maintaining separate codebases.

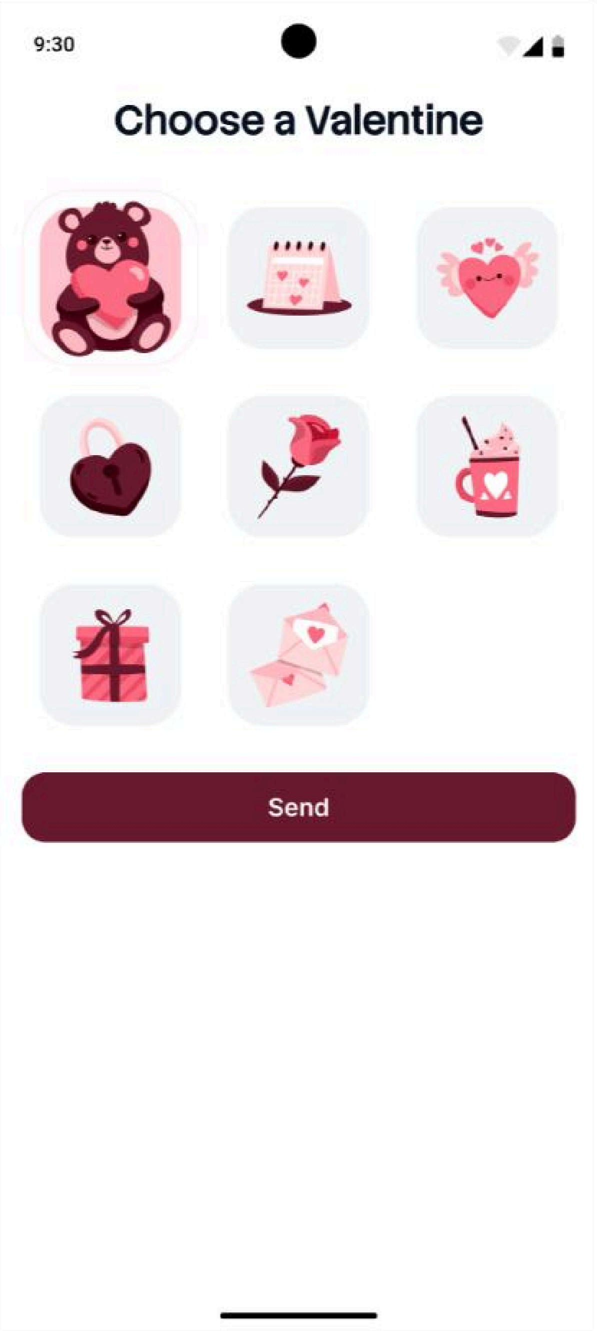
🅰️ Application A — Valentine Sender

Purpose

Valentine Sender is an app that allows the user to select one of the predefined Valentine cards and send it to another app using the standard Android share flow.

Screen Structure

- The app contains **a single screen**.
- The screen is used to:
  - select a Valentine
  - send the selected Valentine



Top Bar

- Positioned at the top of the screen.
- Displays the title: *“Choose a Valentine”*

Valentine Grid

- The main content of the screen is a grid of **8 Valentine cards**.
- Each Valentine:
  - is displayed as an equally sized card
  - contains a Valentine illustration
- All illustrations must match the provided design mockups.

Selection Behavior

- The user can select one Valentine.
- The selected Valentine has a clearly visible selected state.
- Tapping the same Valentine again removes the selection.

Send Button

- The Send button is placed at the bottom of the screen.
- Button states:
  - **Disabled** — when no Valentine is selected
  - **Enabled** — when a Valentine is selected
- The visual style of the button changes according to its state.

Send Action

- When the Send button is tapped:
  - the system **Android Share Chooser** is opened
  - the selected Valentine is made available for sharing
- The user selects the receiving app from the system chooser.

🅱️ Application B — Valentine Receiver

Purpose

Valentine Receiver is an app that receives a Valentine card through the system Android share flow, displays it to the user in a preview mode, and allows the user to either save or dismiss the received Valentine.

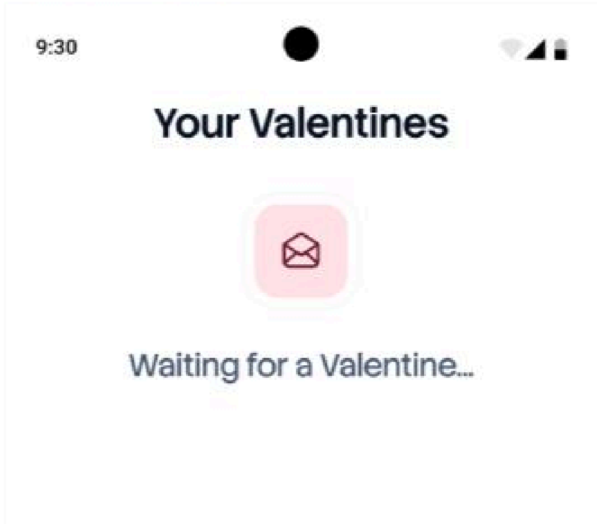
Screen Structure

- The app uses **a single screen** that changes based on the current state.
- No in-app navigation is required.
- All states are implemented as variations of the same screen.

Top Bar

- Positioned at the top of the screen.
- Displays the title: *“Your Valentines”*
- No additional actions or buttons are present.

Receiver States



The app has three main states:

State 1 — Empty / Waiting

This state is shown when:

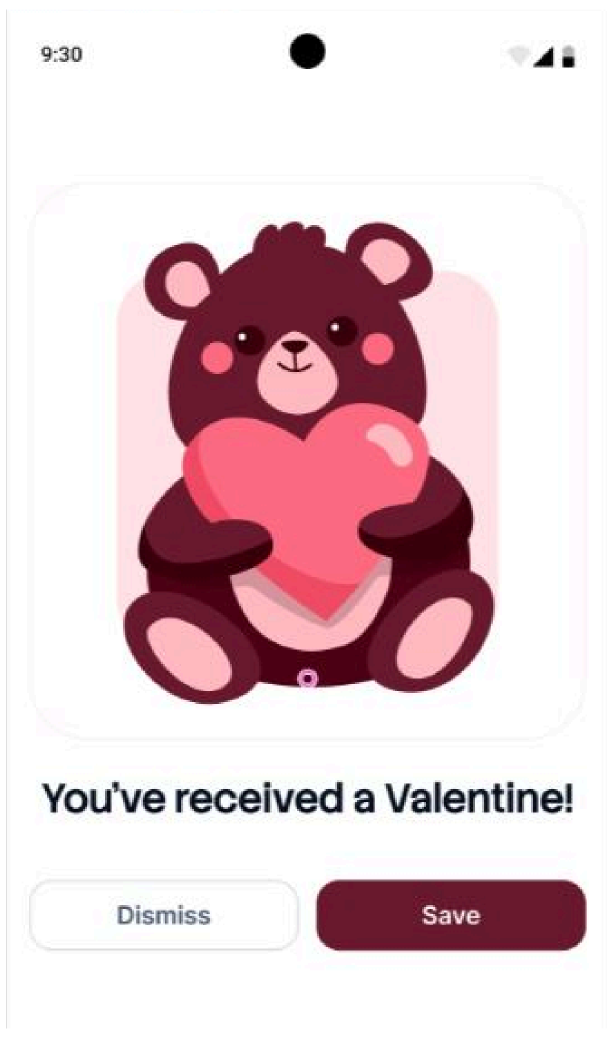
- the app is opened for the first time
- no Valentine has been received or saved yet

UI elements:

- a central icon or illustration related to waiting
- status text: *“Waiting for a Valentine…”*

State 2 — Preview (Receiving)

This state is shown every time a new Valentine is received via a share intent.



You've received a Valentine!

UI elements:

- a large preview of the received Valentine image
- title: *“You’ve received a Valentine!”*
- two action buttons:
  - **Save**
    - saves the received Valentine
    - navigates to the Gallery state after saving
  - **Dismiss**
    - discards the received Valentine
    - does not save it
    - returns the app to the previous state (Empty or Gallery, depending on whether saved Valentines exist)

Behavior:

- this state has **the highest priority** and overrides all other states
- the user’s attention is fully focused on the received Valentine

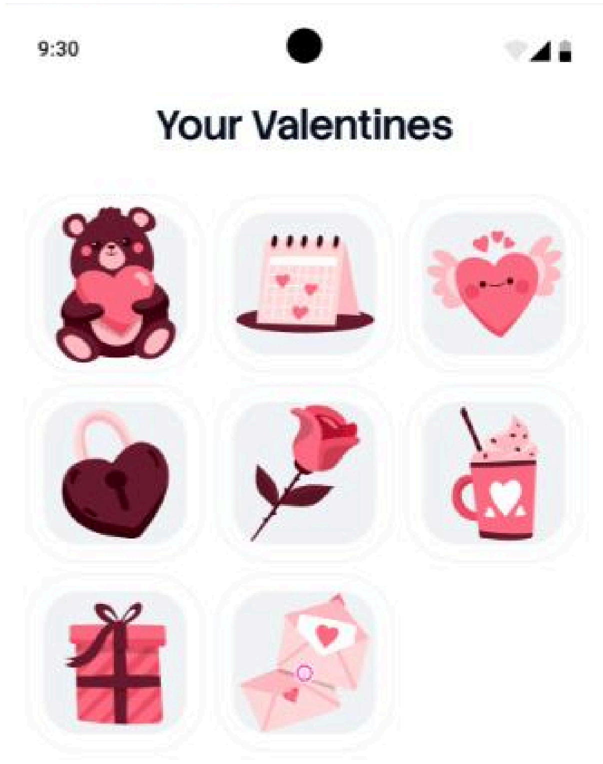
State 3 — Gallery

This state is shown when:

- the user has saved **at least one** Valentine

UI elements:

- a grid gallery of all saved Valentines
- all gallery items are displayed as equal-sized elements
- the screen title remains: *“Your Valentines”*



Your Valentines

🤔 What's Allowed?

- Standard Android/Jetpack libraries
- No 3rd party libraries are allowed or would be required to complete this challenge
- Using standard Android intents to receive shared images from other apps.

⚠️ What's not important

- Responsiveness across every device size or orientation is not mandatory.
- Light / Dark mode.
- Advanced image editing or manipulation.
- Complex navigation or multi-screen flows.
- Animations or visual effects.
- Handling all possible edge cases related to external file sharing.

🔗 Useful Links for This Challenge

- Intents & Intent Filters
- FileProvider
- Intents and intent filters
- Receive simple data from other apps
- Configure product flavors
- 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 **200 XP**.
- Your submission must include:
  a. A **Gist link** with your implementation.
  b. A **screen recording** (max 30 seconds) showing:
    - Launching the **Valentine Sender** app.
    - Selecting a Valentine card.
    - Tapping **Send** and opening the system Android share chooser.
    - Selecting the **Valentine Receiver** app from the chooser.
    - Showing the Preview (Receiving) state with the received Valentine and tapping Dismiss.
    - Sending another Valentine from the Sender.
    - Showing the Preview state again.
    - Tapping **Save** and confirming the Valentine appears in the Gallery.