Technical Requirements – Holiday Cashback Activation

Challenge is accessible on Memberspot: https://pl-coding.mymemberspot.io/library/jx3b7Qik9ip5qpNI8IF2/CAxRTS5yWwDcZ19gwHz0/mkKZb62DwEY9q3Hf9XGG/details

🎭 Scenario

The user enters their bank card number to activate a special holiday cashback bonus.

🎨 Figma Mockups

> https://www.figma.com/design/lLcf8BTbExJf1drvKksZ2O/Winter-Magic-Series?node-id=75-348

🔤 Font - Montserrat

🎯 Feature Goal

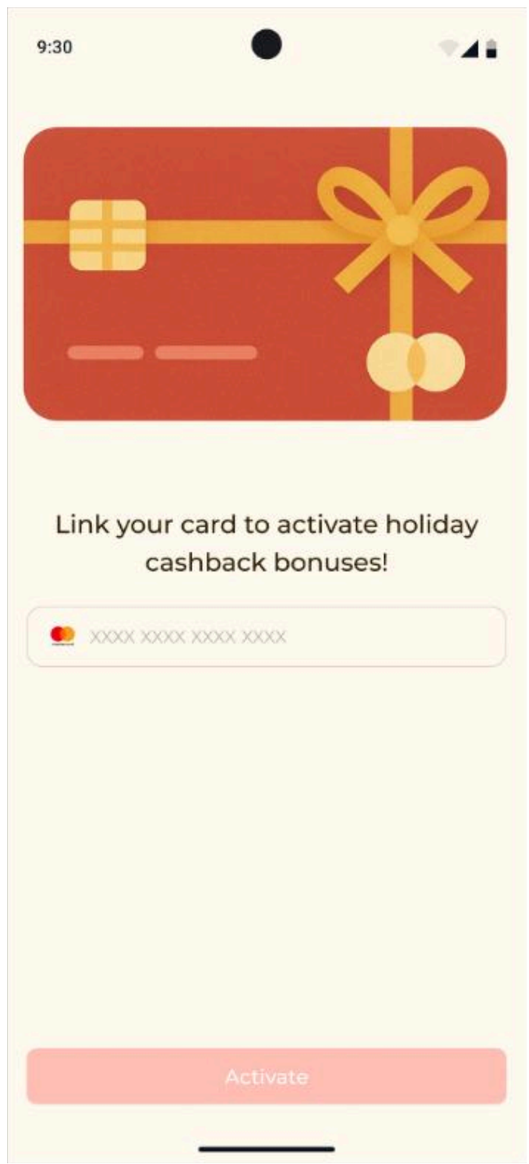
Implement a card input screen with automatic formatting, Luhn validation, visual feedback states, and a final confirmation message.

The goal is to practice building a real-world payment field with input sanitization, masking, and dynamic state handling.

📌 Requirements

Card Illustration

- Located at the top of the screen.
- A decorative festive card illustration with a ribbon.
- Static, non-interactive element.



Title & Description

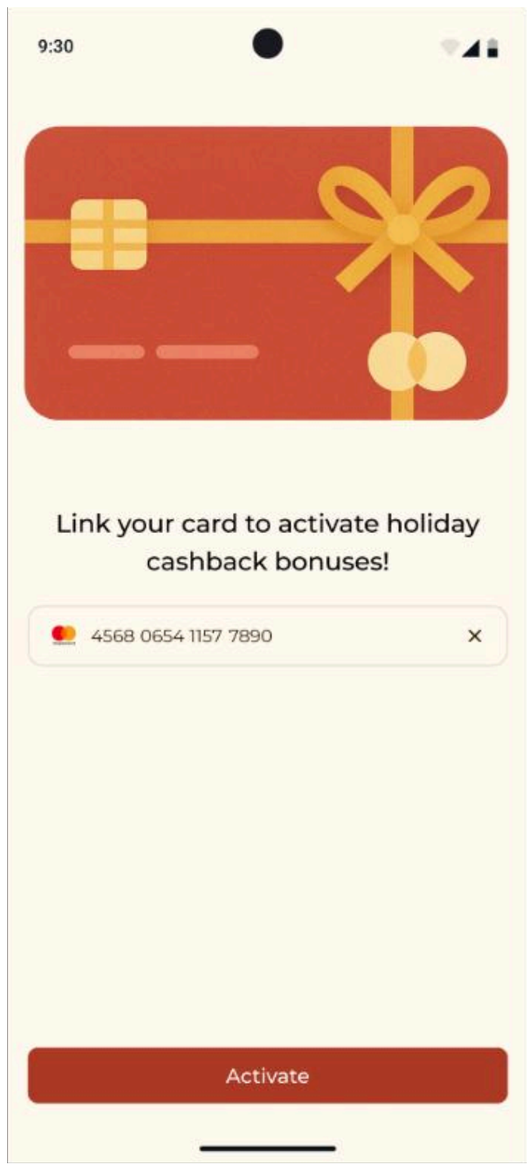
- Placed under the illustration.
- Text: *“Link your card to activate holiday cashback bonuses!”*
- Centered horizontally.

Card Number Field

The main functional element responsible for input, formatting, and validation.

Input Formatting

- Input format: #### #### #### ####.
- The user can enter **digits only**.
- All non-numeric characters (spaces, dashes, letters) are automatically removed.
- A space is automatically inserted **after every 4 digits** to improve readability.
  - Example: entering 4568065411577890 automatically displays as 4568 0654 1157 7890.
- The total number of digits (excluding spaces) is **16**.



Card Type Detection

- Card type is automatically identified:
  - Starts with **4 → Visa** icon appears.
  - Starts with **2 or 5 → MasterCard** icon appears.
- If no number is entered yet, **MasterCard** is displayed by **default**.
- The icon appears inside the left side of the TextField.

Clear Icon

- A × icon appears on the right side as soon as typing begins.
- Tapping it **clears the entire input field**.

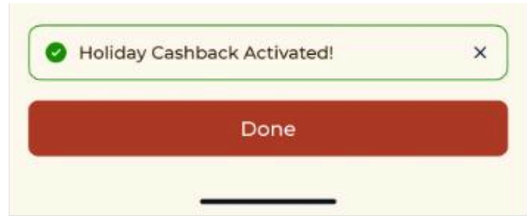
Luhn Validation

- Once **16 digits** are entered, perform a Luhn check to verify if the card number is valid.
- How to validate using Luhn algorithm (simple formula):
  a. Starting from the rightmost digit, move left.
  b. Double every second digit.
  c. If the result > 9, subtract 9 from it.
  d. Sum all digits.
  e. If the total sum % 10 == 0 → the card number is valid.
- If valid:
  - The border remains neutral.
  - The **Activate** button becomes enabled.
- If invalid:
  - The border turns **red**.
  - The helper text *“Invalid card number”* appears below the field.
  - The Activate button stays disabled.
  - If the user edits the number after an error, the field **returns to normal state** (neutral border, no message). The validation runs again only when all **16 digits** are entered.

Activate Button

- Initially **disabled**.
- Becomes **enabled** only after a valid 16-digit card number passes the Luhn check.

- When pressed:



1. Enters a “processing” state — shows text *“Activating…”* and a **CircularProgressIndicator**.
2. The progress indicator runs for exactly **2 seconds** (hardcoded delay).
3. After 2 seconds, a **Snackbar** appears with the message: *“Holiday Cashback Activated!”*
4. The button text changes to **Done** and becomes active.
5. When the user presses **Done**, the screen resets:

- The input field is cleared.
- The **Activate** button returns to its initial disabled state.

🤔 What's Allowed?

- Standard Android / Jetpack libraries.
- Use Material 3 components or simple custom UI elements.
- No 3rd party libraries are allowed or would be required to complete this challenge

⚠️ What's not important

- Responsiveness across every device size or orientation is not mandatory.
- Actual card linking or network communication.
- State persistence after closing the screen.
- Light/Dark mode support.

🔗 Useful Links for This Challenge

- Luhn algorithm
- Luhn algorithm Wiki
- Visual Transformation
- Configure text fields
- Create a notification with a snackbar
- 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 20 seconds) showing:
    - Initial state (empty field, disabled button).

- Typing or pasting a random invalid number → shows red border + *“Invalid card number”*.
- Entering the **valid test card number** 4242 4242 4242 4242 → button activates.
- Press **Activate** → 2s loading → *“Holiday Cashback Activated!”* Snackbar → button text changes to **Done**.
- Press **Done** → all fields reset to the initial state.