Technical Requirements – Holiday Movie Collection

Challenge is accessible on Memberspot: https://pl-coding.mymemberspot.io/library/jx3b7Qik9ip5qpNI8IF2/BwAxp75QNXGaDGF2ZJEG/v7G6MS5eKEUZhr9ZOzbB/details

🎭 Scenario

This mini-challenge simulates a simple holiday movie collections app where users create custom bundles of movies. The challenge focuses on practicing local data persistence using Room, with an emphasis on relational database design and many-to-many relationships between movie bundles and movies.

🎨 Figma Mockups

> https://www.figma.com/design/XKuvJCxCybCUpKZP0LelAs/New-Year-Fresh-Start?node-id=1-3

🔤 Font - Plus Jakarta Sans

🎯 Feature Goal

Practice using Room to model relational data, including a many-to-many relationship, while building a simple multi-screen app with basic navigation and list-based UI.

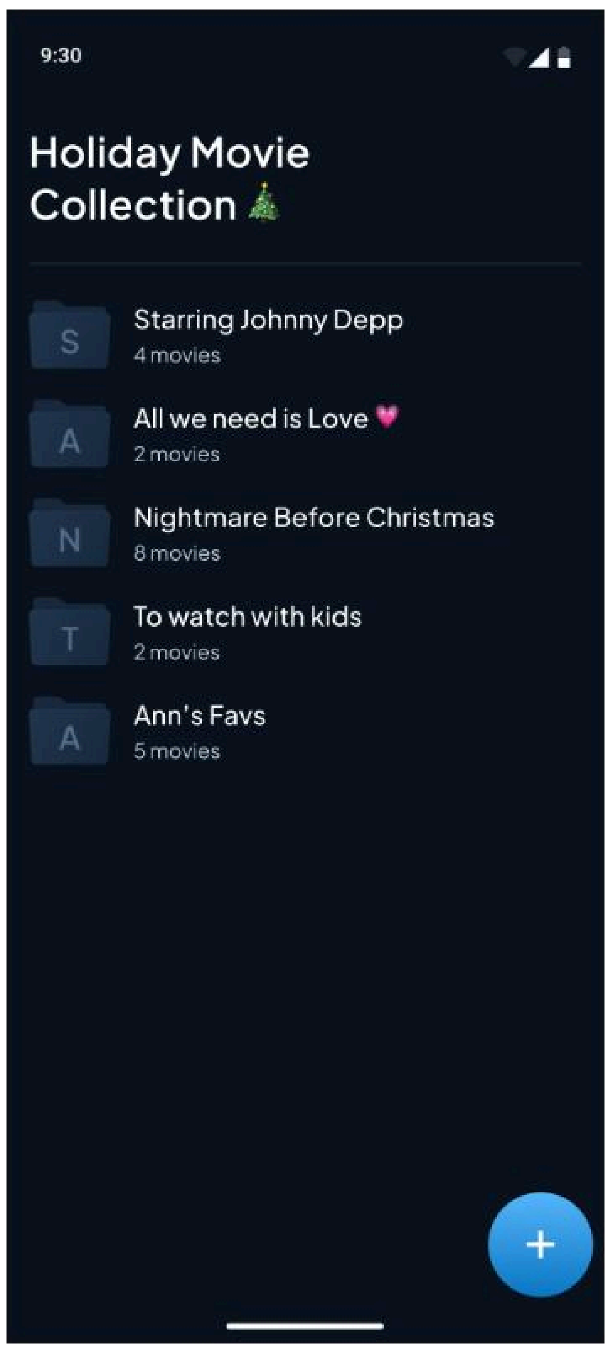
📌 Requirements

Main Screen (Movie Bundles List)

The main screen displays a list of holiday movie collections created by the user. This screen serves as the entry point of the app and provides access to existing collections as well as the option to create new ones.

Top Bar

- Positioned at the top of the screen.
- Displays the title *“Holiday Movie Collection”*.
- Contains **no navigation** actions or buttons.



Movie Bundles List

- Positioned below the Top Bar.
- Implemented as a vertical list.
- Displays all movie bundles created by the user.
- The list scrolls vertically when the content exceeds the screen height.

Movie Bundle Item

Each list item represents a single movie bundle and includes:

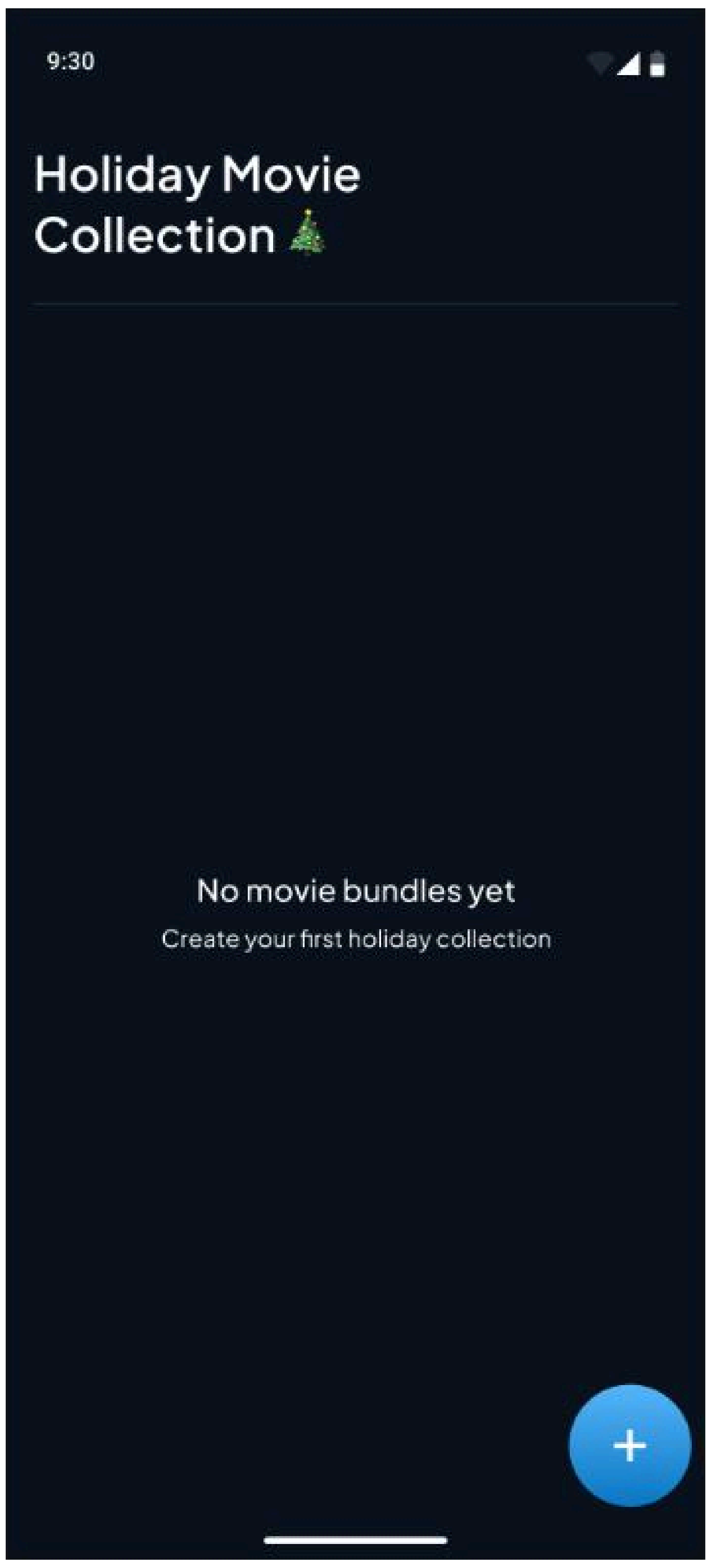
- The bundle name.
- A subtitle showing the number of movies in the bundle (for example, *“2 movies”*, *“5 movies”*).
- A folder-style icon displaying the **first letter** of the bundle name.
- The entire item area is **clickable**.
- Tapping a bundle item opens the details screen for the selected bundle.

Create Bundle Button

- Positioned at the bottom of the screen.
- Implemented as a Floating Action Button with a “+” icon.
- The button is fixed and does not scroll with the list.
- Tapping the button opens the **Create Bundle** screen.

Empty State

- Displayed when no movie bundles have been created.
- Centered vertically on the screen.
- Consists of two text elements:
  - a primary message: *“No movie bundles yet”*;
  - a secondary message: *“Create your first holiday collection”*.
- The empty state does not include any additional actions or buttons.
- The Floating Action Button remains visible and is used to create the first bundle.



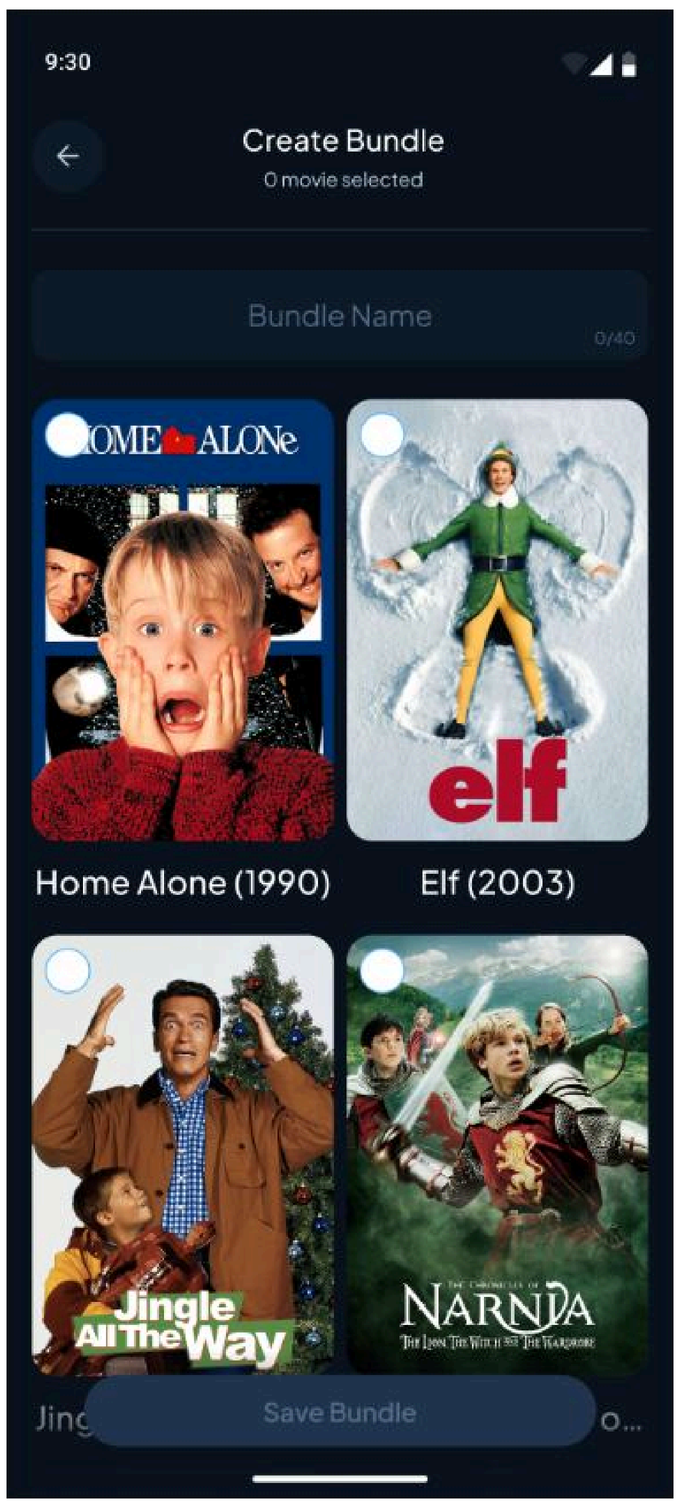
Holiday Movie Collection 🎄

No movie bundles yet

Create your first holiday collection

Create Bundle Screen

This screen is used to create a new holiday movie collection. The user enters a bundle name, selects movies from a predefined list, and saves the created bundle.



Top Bar

- Positioned at the top of the screen.
- Contains a **Back button**.
- Displays the title *“Create Bundle”*.
- Displays a subtitle showing the number of selected movies (for example, *“1 movie selected”*).
- The subtitle updates dynamically when the selection changes.

Bundle Name Field

- Positioned below the Top Bar.
- Implemented as a text field.
- Used to enter the bundle name.
- Displays the placeholder text *“Bundle Name”*.
- Has a **maximum character limit** for 40 characters.
- Input is restricted to 40 characters; additional characters cannot be entered.
- The current character count is displayed inside the text field, in the bottom-right corner.
- The field remains fixed and does not scroll with the content.

Movie Poster Item

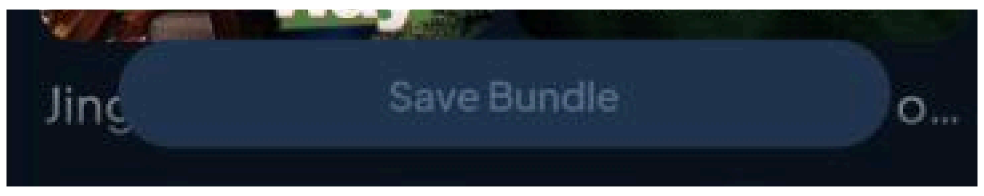
Each grid item represents a single movie and includes:

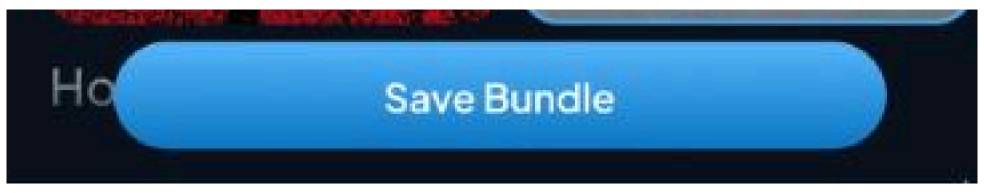
- A movie poster.
- The movie title with release year displayed below the poster.
- A checkbox or selection indicator displayed on the poster.

Selection Behavior

- Tapping a poster toggles its selection state.
- Selected movies are visually highlighted.
- Tapping a selected poster again removes the selection.
- The number of selected movies is reflected immediately in the Top Bar subtitle.

Save Bundle Button



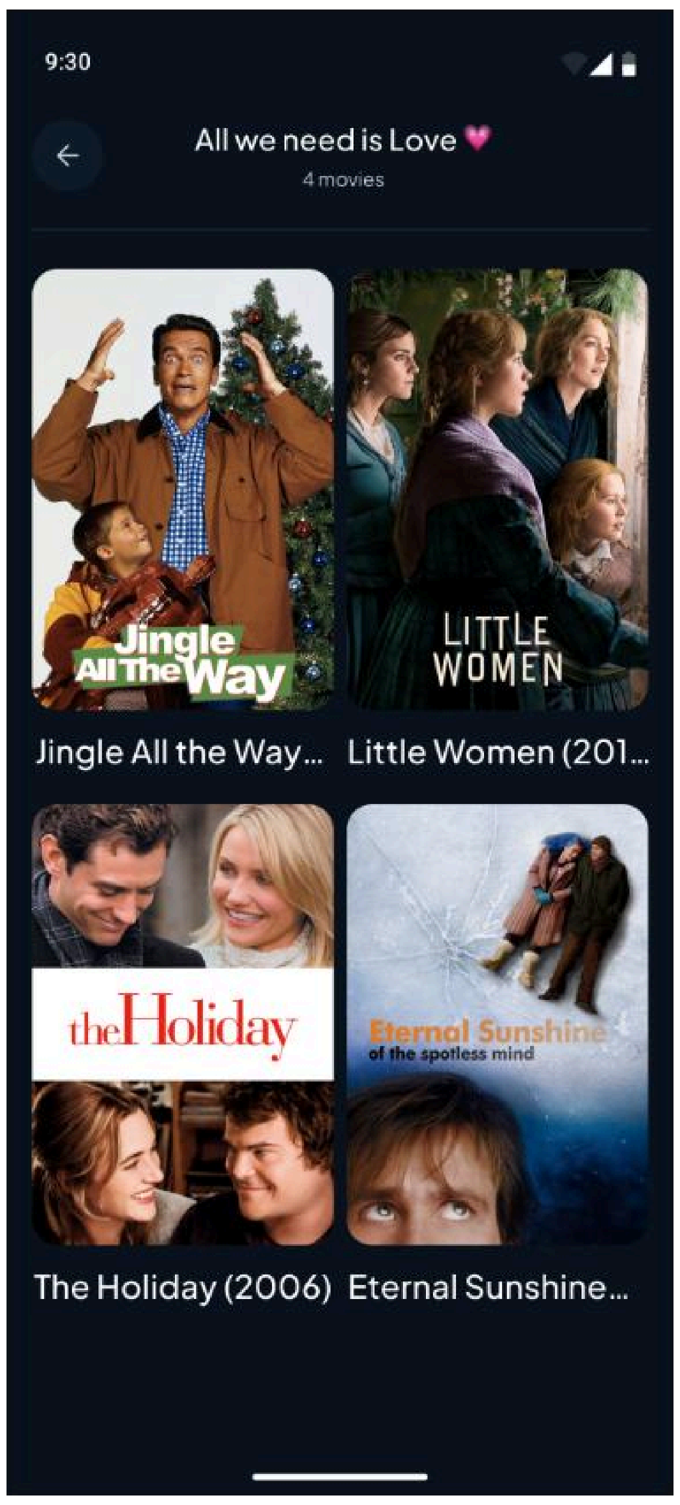


- Positioned at the bottom of the screen.
- Fixed and does not scroll with the grid.
- Displays the label “Save Bundle”.
- The button is visible at all times.
- The button is displayed in a disabled state until:
  - a bundle name is entered;
  - at least one movie is selected.
- When enabled and tapped:
  - the new bundle is saved;
  - the user is navigated back to the main screen showing the list of bundles.

Bundle Details Screen

This screen displays the contents of a selected holiday movie bundle. It is opened when the user taps a bundle item on the main screen.

Top Bar



- Positioned at the top of the screen.
- Contains a **Back button**.
- Displays the bundle name.
- Shows a subtitle with the number of movies in the bundle (for example, *“4 movies”*).
- Pressing the Back button:
  - navigates the user back to the main screen;
  - removes the Bundle Details screen from the navigation back stack.

Movies Grid

- Positioned below the Top Bar.
- Implemented as a two-column grid.
- Displays only the movies that belong to the selected bundle.
- The grid scrolls vertically if the number of movies exceeds the screen height.

Movie Poster Item

Each grid item represents a single movie and includes:

- A movie poster.
- The movie title with release year displayed below the poster.

🧪 Initial Movie List

The app should start with a predefined list of movies that is available on the Create Bundle screen. This list represents the base set of movies from which users can create their custom bundles.

The following movies should be included:

- Home Alone (1990)
- Elf (2003)
- Jingle All the Way (1996)
- The Chronicles of Narnia: The Lion, the Witch and the Wardrobe (2005)
- The Polar Express (2004)
- Klaus (2019)
- Little Women (2019)
- The Holiday (2006)
- Eternal Sunshine of the Spotless Mind (2004)
- Krampus (2015)
- The Thing (1982)
- Die Hard 2 (1990)
- Wind River (2017)

Poster images for these movies can be found in the attached file on the members area.

🤔 What's Allowed?

- Standard Android/Jetpack libraries
- Room for local data persistence.
- Simple navigation between screens.

⚠️ What's not important

- Responsiveness across every device size or orientation is not mandatory.
- Light/Dark mode support.
- Advanced error handling or edge cases.
- Editing or deleting existing bundles.
- Animations or visual effects.

🔗 Useful Links for This Challenge

- Room Persistence Library
- Define relationships in Room
- Persist data with Room codelab
- Lists and grids
- Text Fields - UX With Material3
- Navigation3 Basics
- 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 40 seconds) showing:
    - Opening the app and displaying the main screen with the list of movie bundles.
    - Creating a new bundle by:
      - entering a bundle name;
      - selecting one or more movies;
      - saving the bundle.
    - Returning to the main screen and confirming that the new bundle appears alongside the existing ones.
    - Opening different bundles (including the newly created one) to show:
      - the correct list of movies for each bundle;
      - that each bundle displays its own related movies correctly.
    - Restarting or reopening the app and confirming that:
      - all bundles are still present;
      - movie-to-bundle relationships are preserved.