

CLOUDCARDS

The **Cloudcards** is designed to help users memorize different topics through interactive flashcards. Each flashcard contains a question and its corresponding answer. The app features two learning modes:

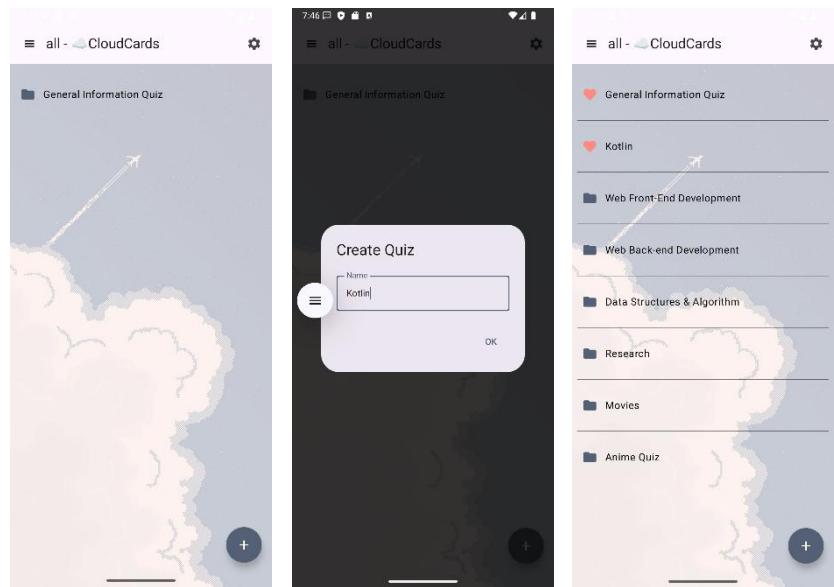
- Training Mode: Presents a question and let users think the answer
- True/False Mode: Presents a statement that the user must judge as true or false

The app uses Room for local data persistence, allowing users to create, edit, and organize their flashcards while tracking their learning progress.

7 Screenshots of the Application

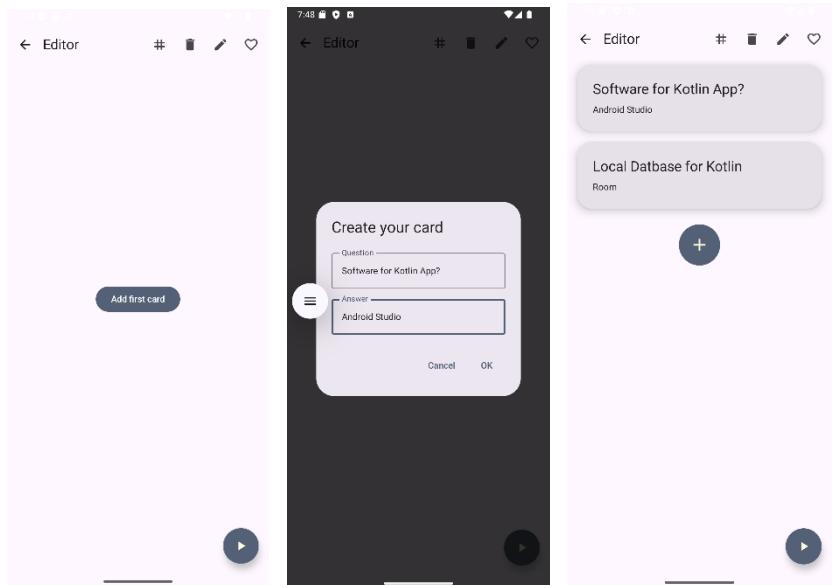
Screenshot 1: Main Screen

- Shows list of all flashcards
- Floating action button to add new quizzes



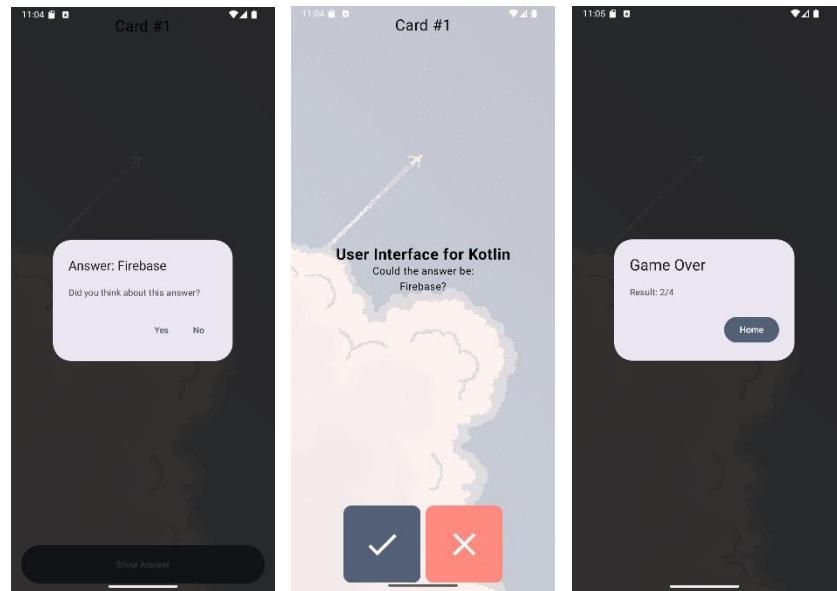
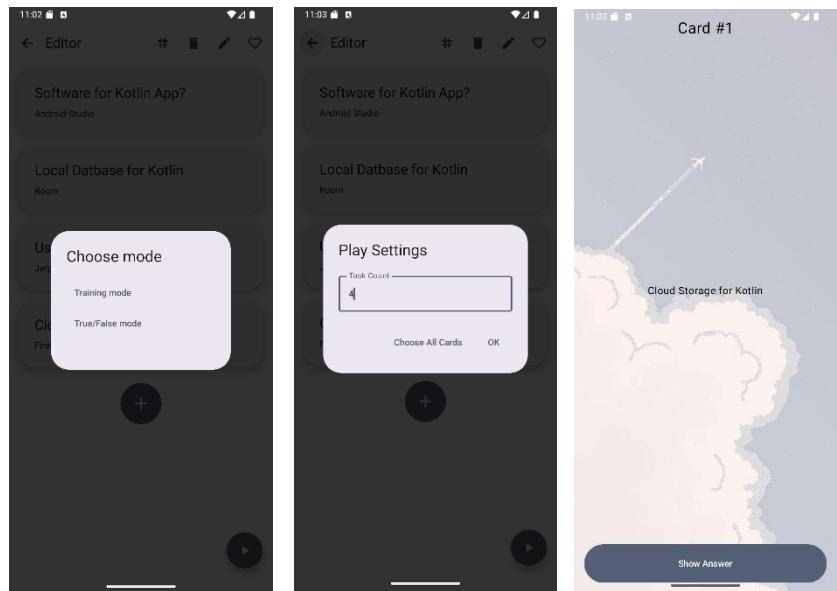
Screenshot 2: Card Creation Screen

- Floating action button to add new cards
- Form with fields for question and answer
- Save/cancel buttons
- Floating action button to play the quiz



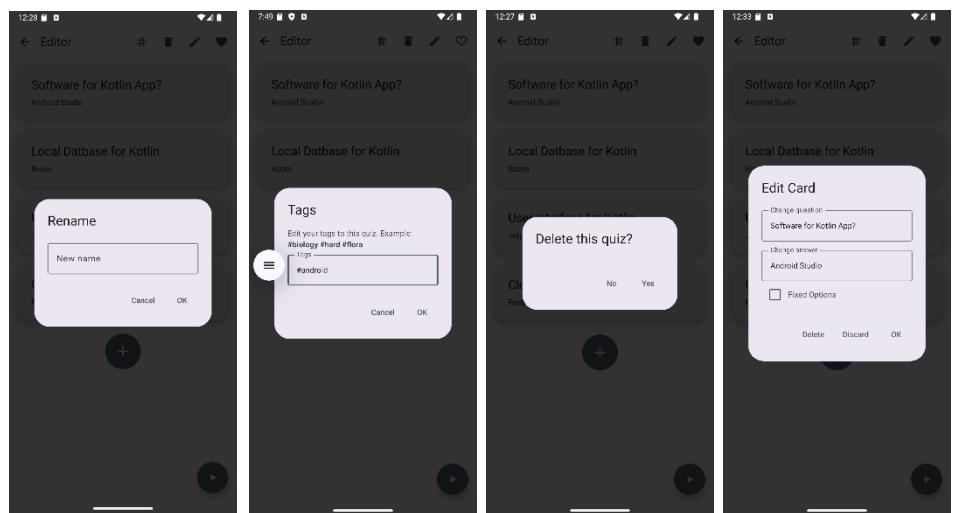
Screenshot 3: Play Game Quiz

- Play quiz with game option of training and true or false mode
- Input dialog for the number of tasks to be played or choose all cards
- Displays current question
- Floating action button for training mode to show the current answer
- True/False buttons
- Game Score at the end



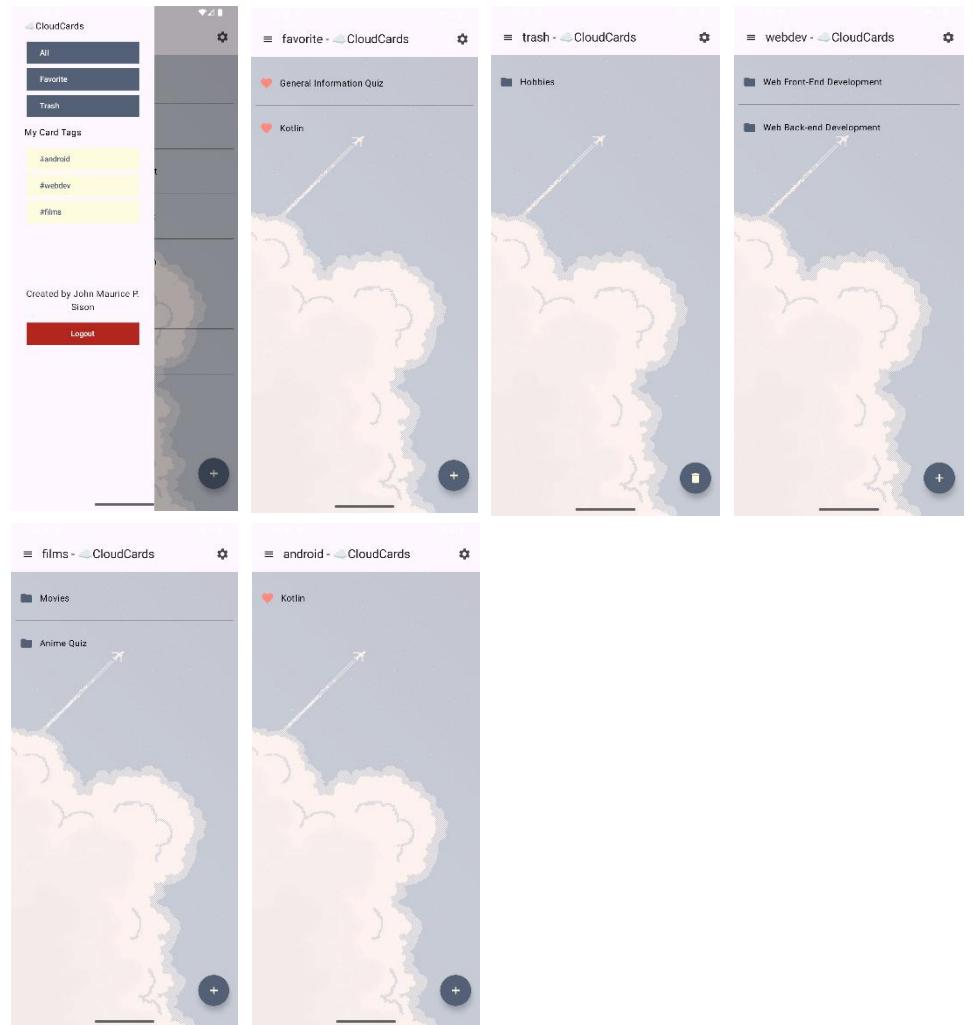
Screenshot 4: Quiz/Card Editor

- Top Bar with buttons to edit the quiz
- Add tags (e.g. #androidcourse) in quizzes
- Delete quiz
- Rename quiz
- Favorite quiz
- Edit Card
- Delete Card



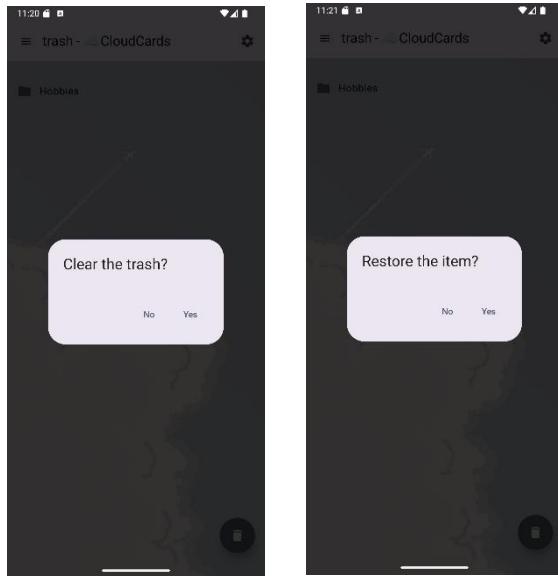
Screenshot 5: Sidebar

- Organize quizzes through buttons of show all cards, favorites, card tags, and recently deleted.
- Logout button



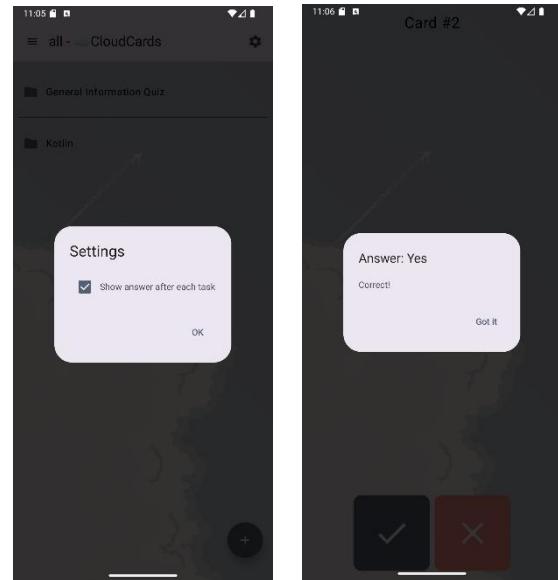
Screenshot 6: Clear and Restore Trash

- Floating action button to clear the trash
- Card can be restored



Screenshot 7: App Settings

- Checkbox option to show the answer after each task in True or False mode.



CloudCards Project Files/Source Code with Description

Dependencies

```
61     implementation(libs.androidx.room.runtime) //Room runtime for database operations
62     implementation(libs.androidx.room.ktx) //Kotlin extensions (coroutines support) for Room.
63     ksp(libs.androidx.room.compiler) //Room annotation processor for generating DAO & entity code
64
65     implementation(libs.androidx.lifecycle.viewmodel.compose) //Allows using ViewModel directly in Compose
66
67     implementation(libs.androidx.navigation.compose) //Enables in-app navigation with Compose
68     implementation(libs.androidx.material.icons.extended) //Provides Material Icons for Compose
69     implementation(libs.ui) //Jetpack Compose UI core components.
70     implementation(libs.accompanist.systemuicontroller) //Allows controlling system UI (status bar, nav bar) in Compose
71 }
```

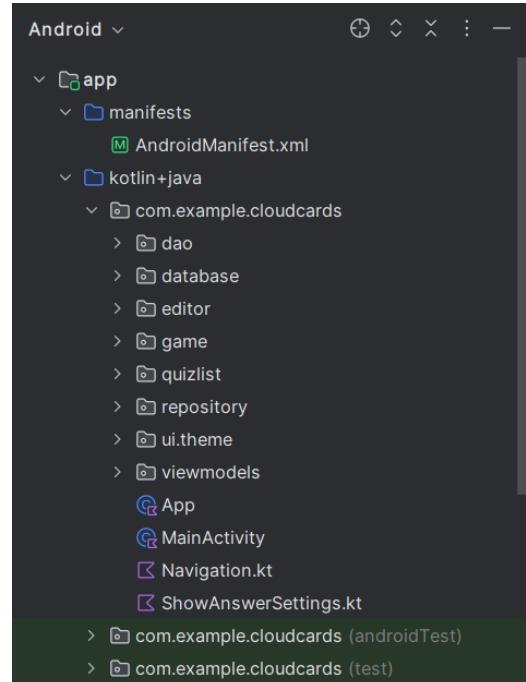
The project was organized with packages such as:

- DAO
- Database
- Editor (Dialogs and Screen for Editing)
- Game (Dialogs and Screen for Game Mode)
- Quizlist (Dialogs and Screen for Flashcards/Main)
- Repository
- Ui.theme
- Viewmodels

MainActivity.kt

Shows the app's screen and theme

- It sets up the background image, system bar color, and theme, then starts NavApp with the app's pages can show quizzes and cards using the data from the database.



Navigation.kt

Sets up screen navigation for the app using Jetpack Compose Navigation

- It defines routes with parameters, uses NavHost + navController to switch screens while keeping the back stack, and passes my ViewModels so screens can show and update quiz data.

App.kt

Sets up the Room database when the app starts

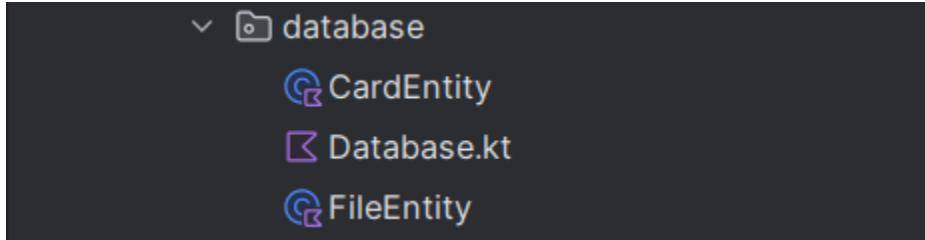
- It migrates safely, inserts a demo “General Information Quiz” on first launch using coroutines to avoid UI lag, and flags first launch as complete to prevent duplicate data.

ShowAnswerSettings.kt

Manages the app's "show answer" setting using Shared Preferences

- It wraps preferences in PrefViewModel with StateFlow for reactive Compose UI updates, using PrefViewModelFactory for clean MVVM injection.

Database package



FileEntity.kt

Stores each **quiz file** in the files table. It keeps the **name, tags, favorite status, and trash status** of the quiz.

CardEntity.kt

Stores each flashcard in the cards table. It links to the quiz it belongs to using fileId (cards auto-delete if the quiz is deleted).

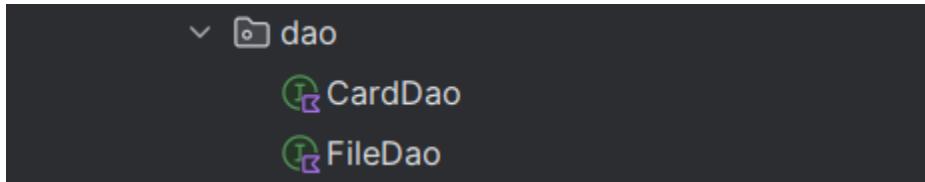
- It saves the question, answer, and optional multiple-choice options, using auto-increment IDs and indexing for faster lookups

Database.kt

Defines the Room database for the app using CardEntity and FileEntity as tables (version 3 for migrations)

- Provides cardDao() and fileDao() to manage cards and quiz files.
- Room uses this to auto-generate the SQL and DAO code, so raw SQL is not needed

DAO package



CardDao.kt

Interface for Room to insert, update, delete, and fetch cards in the database.

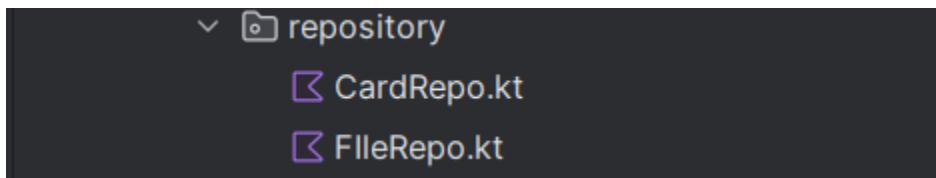
- Uses suspend functions so operations run in coroutines without blocking the UI.
- `getByFileId` retrieves cards linked to a specific quiz file

FileDao.kt

Interface for Room to manage quiz files (`FileEntity`) in the database.

- Supports insert, update, delete, and fetch operations with suspend for coroutine safety.
- Uses `OnConflictStrategy.REPLACE` to replace files cleanly if conflicts happen, keeping IDs consistent.

Repository package

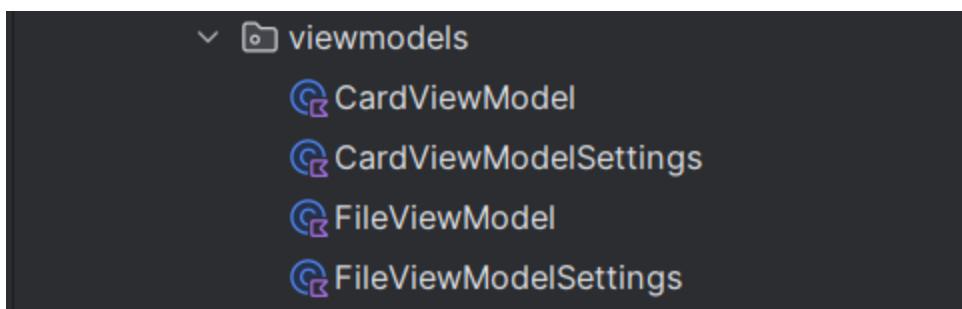


Act as a middle layer between your DAOs and ViewModels by managing how the app gets data from the database cleanly and safely.

CardRepo.kt: Handles all card data operations cleanly between the ViewModel and Room database.

FileRepo.kt: Manages quiz file data operations cleanly between the ViewModel and Room database.

Viewmodels package



Contains the app's ViewModel classes used in CloudCards and connects the app's data to the app's UI in a clean, reactive, and organized way.

CardViewModel.kt

Manages card data for the UI using StateFlow and coroutines, allowing the app to add, update, delete, and fetch cards safely.

CardViewModelSettings.kt

Creates CardViewModel with CardRepository injected for dependency management during ViewModel initialization.

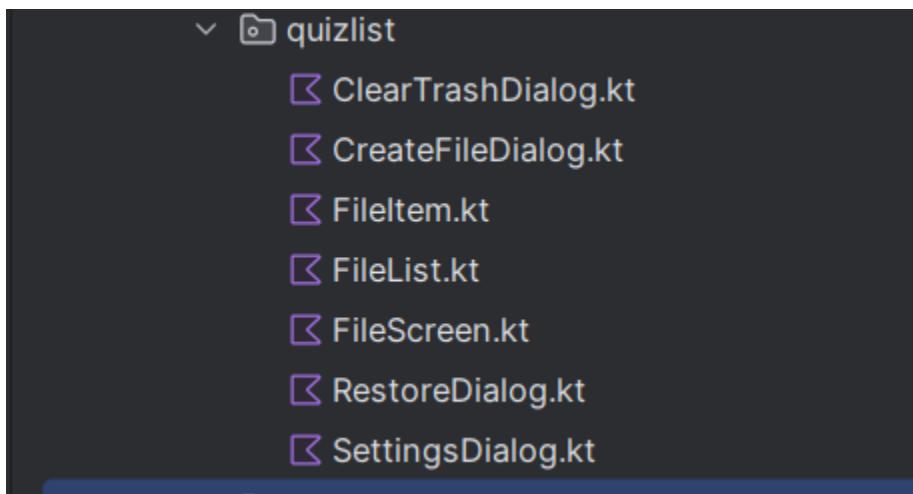
FileViewModel.kt

Manages quiz file data for the UI using StateFlow, providing filtering, trash management, favorites, tagging, and CRUD operations safely.

FileViewModelSettings.kt

Creates FileViewModel with FileRepository injected, allowing safe dependency management when initializing the ViewModel in the app.

Quizlist package



Holds all Compose UI screens and reusable components for listing, creating, managing, and displaying quiz files in CloudCards.

- It connects the FileViewModel and CardViewModel to the UI so users can view, add, rename, delete, and filter quiz files easily.

FileItem.kt: Shows a single quiz file with icon and name, clickable to open.

FileList.kt: Displays a scrollable list of quiz files using FileItem.

FileScreen.kt: Main screen to view, filter, add, delete, and manage quiz files.

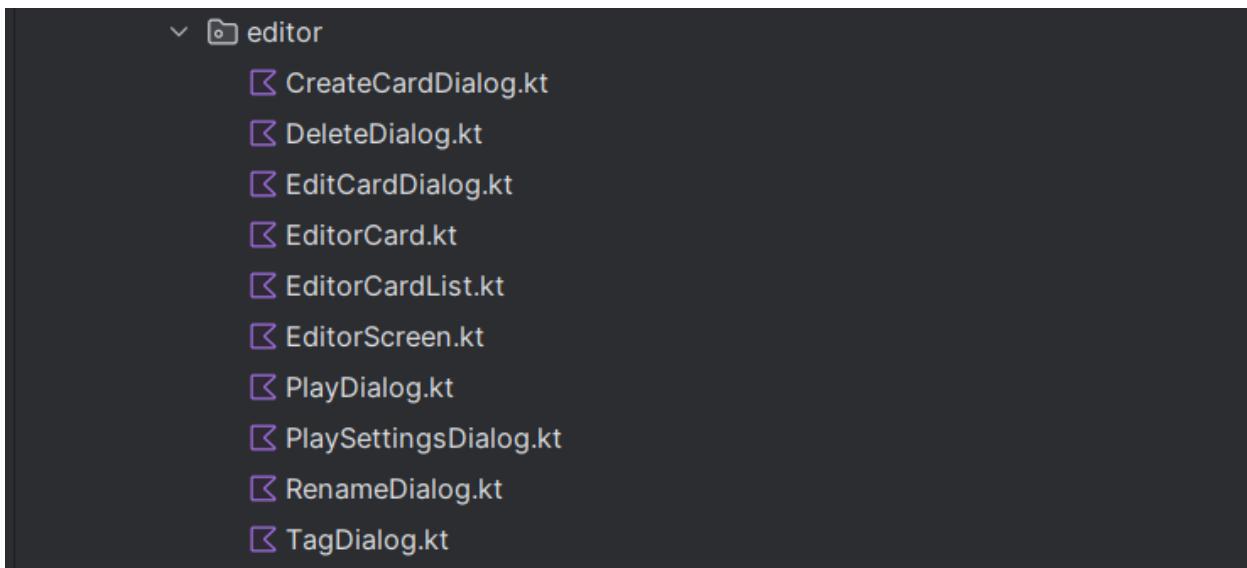
CreateFileDialog.kt: Dialog for creating a new quiz file by entering its name.

SettingsDialog.kt: Dialog to toggle the “show answer after each task” setting.

ClearTrashDialog.kt: Dialog to confirm clearing all trashed quiz files permanently.

RestoreDialog.kt: Dialog to confirm restoring a quiz file from trash.

Editor package



Holds screens and dialogs for viewing, adding, editing, deleting, and playing flashcards within a specific quiz file in CloudCards.

- It lets users manage cards inside a quiz and start quiz modes cleanly, keeping card management organized and separated from file management in the app structure.

CreateCardDialog.kt: Dialog to add a new flashcard with a question and answer.

DeleteDialog.kt: Dialog to confirm deleting a quiz file (moves it to trash).

EditCardDialog.kt: Dialog to edit an existing flashcard’s question, answer, and options.

EditorCard.kt: UI component showing a single flashcard in the editor.

EditorCardList.kt: Shows a scrollable list of flashcards for a quiz file.

EditorScreen.kt: Main screen for editing a quiz file: add, edit, delete, tag, or play cards.

PlayDialog.kt: Dialog for choosing how to play a quiz (mode selection).

PlaySettingsDialog.kt: Dialog for selecting play settings, like how many cards to use.

RenameDialog.kt: Dialog to rename a quiz file.

TagDialog.kt: Dialog to add or edit tags for a quiz file for filtering.

Comple Source Code File

https://drive.google.com/drive/folders/1tE58CyK2aogNOzvP48cC_2owBPlukW_8?usp=sharing