



LIBRARY APP
GRUPPO 20

MOBILE PROGRAMMING A.A 2024/2025
Crisci Chiara
Crivo Luisa
De Pascale Francesca
Del Sorbo Palma
Graziuso Emanuela

SOMMARIO

1. INTRODUZIONE.....	2
1.1 Presentazione del progetto e contesto.....	2
1.2 Obiettivo dell'applicazione.....	2
2. METODOLOGIA.....	3
2.1 Concettualizzazione e Definizione.....	3
2.2 Design e Prototipazione.....	3
2.3 Sviluppo e Integrazione.....	3
3. DESIGN NEL DETTAGLIO.....	4
3.1 DIAGRAMMA MIRO.....	5
4. STRUTTURA DEL PROGETTO FLUTTER.....	6
5. SCELTE PROGETTUALI.....	7
5.1 Framework e architettura.....	7
5.2 Persistenza dei dati.....	7
5.3 Consistenza e integrità dei dati.....	7
5.4 Esperienza utente e responsività.....	7
5.5 Gestione delle immagini.....	7
5.6 Navigazione tra schermate.....	8
5.7 Versionamento e collaborazione.....	8
5.8 Gestione dei pacchetti.....	8
6. DESCRIZIONE DETTAGLIATA DELL'APPLICAZIONE.....	8
6.1 Schermata principale (Home).....	9
6.2 Schermata aggiunta/modifica libro.....	10
6.3 Schermata dettaglio libro.....	11
6.4 Schermata categoria.....	12
6.5 Schermata dettaglio categoria.....	12
6.6 Schermata analisi (Activity).....	13
6.7 Schermata ricerca.....	14

1. INTRODUZIONE

1.1 Presentazione del progetto e contesto

Library App è un'app mobile multiplataforma sviluppata in Flutter, pensata per offrire agli utenti un sistema moderno, intuitivo e personalizzabile per la gestione della propria libreria personale. In un'epoca in cui la lettura digitale è sempre più diffusa, l'app risponde all'esigenza di avere uno strumento efficace per organizzare, monitorare e analizzare le proprie abitudini di lettura.

L'app consente di catalogare i libri, seguirne lo stato di lettura, aggiungere valutazioni, note personali e assegnarle a categorie personalizzate. Inoltre, una sezione dedicata all'analisi statistica permette all'utente di riflettere sul proprio percorso di lettura attraverso dati visivi e significativi.

1.2 Obiettivo dell'applicazione

L'obiettivo è sviluppare un'app completa, accessibile e gradevole nell'utilizzo quotidiano, che consenta:

- La creazione e gestione di un archivio personale di libri, distinti per stato di lettura (To Read, Reading, Completed);
- L'assegnazione di valutazioni, commenti e categorie personalizzate;
- La consultazione di analisi statistiche sulle letture effettuate;
- L'esplorazione rapida e organizzata dei contenuti tramite funzioni di ricerca avanzata e filtri dinamici.

2. METODOLOGIA

Lo sviluppo di *Library App* ha seguito un approccio iterativo, articolato in quattro fasi principali: Concettualizzazione, Definizione, Design e Sviluppo.

2.1 Concettualizzazione e Definizione

In fase iniziale sono stati stabiliti:

- **Obiettivi funzionali:** gestione di un archivio personale di libri, assegnazione di categorie, monitoraggio dello stato di lettura (To Read, Reading, Completed), valutazione e commenti, analisi statistica e ricerca avanzata.
- **Target utenti:** lettori che desiderano organizzare la propria librerie e tracciare i progressi di lettura.
- **Funzionalità principali:** definite in base alla coerenza con le finalità del progetto e alla fattibilità tecnica.

2.2 Design e Prototipazione

La progettazione visiva è stata realizzata con Miro, piattaforma per la creazione di wireframe e mockup. Sono stati definiti:

- Layout delle schermate principali
- Posizionamento degli elementi (icone, pulsanti, testi)
- Flussi di navigazione e interazione
- Struttura logica dei contenuti

2.3 Sviluppo e Integrazione

L'app è stata sviluppata con Flutter, framework multipiattaforma basato su un'unica codebase. L'intero processo è stato gestito su GitHub, con sviluppo incrementale e modularizzazione del codice.

Le fasi operative hanno incluso:

- Sviluppo di un MVP con le funzionalità essenziali
- Estensione del progetto con: gestione categorie, filtri avanzati, sistema di preferiti
- Integrazione di SQLite per la persistenza dei dati (libri, categorie, preferiti, stati)
- Ottimizzazione UI/UX per la responsività su smartphone e tablet, in modalità verticale e orizzontale

3. DESIGN NEL DETTAGLIO

Durante le fasi preliminari di progettazione dell'interfaccia utente, è stato realizzato un prototipo navigabile tramite la piattaforma collaborativa Miro, impiegato come strumento di supporto per definire in maniera strutturata l'architettura visiva dell'app *Library App* e i relativi flussi di interazione.

Il prototipo si compone di una serie di **mockup statici** collegati da **linee direzionali codificate per colore**, le quali illustrano in modo intuitivo la **navigazione tra le schermate**. Questa modellazione ha consentito di:

- **Tracciare con precisione i flussi principali di navigazione**, come il passaggio dalla schermata Home al dettaglio di un libro o alla libreria personale;
- **Esplicitare le dipendenze funzionali** tra i componenti, come la correlazione tra l'assegnazione di una categoria e la sua schermata di riepilogo;
- **Mantenere uniformità grafica e comportamentale** grazie all'utilizzo ricorrente di elementi comuni (es. bottom navigation bar, azioni contestuali, pop up interattivi);
- **Simulare scenari dinamici**, quali il processo di aggiunta di un libro e il successivo reindirizzamento alla sua visualizzazione dettagliata.

L'impiego di questo approccio visuale ha permesso una **rilevazione precoce di potenziali criticità** progettuali, soprattutto in merito alla gestione dello stato di lettura, all'implementazione della persistenza locale e al coordinamento tra sezioni dinamiche come *Activity*, *Search* e *Favorites*.

Dal punto di vista tecnico, il prototipo ha fornito un riferimento concreto per la definizione di:

- Una **struttura gerarchica chiara** dell'interfaccia (albero delle viste e accessi secondari);
- L'**organizzazione logica degli elementi UI**, inclusi componenti interattivi quali icone, card, form di input e sistemi di valutazione;
- Le **modalità previste di interazione**, come filtri dinamici, selezioni contestuali, pop-up modali e navigazione multilivello;
- Le **regole di transizione tra schermate**, garantendo coerenza comportamentale e una user experience fluida e prevedibile.

3.1 DIAGRAMMA MIRO



4. STRUTTURA DEL PROGETTO FLUTTER

Per garantire una gestione ordinata, modulare e facilmente manutenibile del codice, il progetto è stato organizzato secondo i principi di separazione delle responsabilità e riutilizzabilità dei componenti. La struttura delle cartelle all'interno della directory lib/ riflette un'architettura pensata per facilitare sia lo sviluppo collaborativo che la scalabilità futura dell'applicazione.

La seguente struttura rappresenta la gerarchia principale all'interno della cartella lib/, corredata da una descrizione sintetica di ciascun file e cartella:

```
lib/
  core/
    theme.dart          # Tema globale dell'applicazione (colori, font, ecc.)

  data/
    database_helper.dart      # Gestione database SQLite: CRUD per libri e categorie
    fake_books.dart          # Libri fintizi caricati in "Popular now"
    fake_categories.dart     # Categorie fintizie predefinite

  models/
    book.dart            # Modello dati per un libro
    category.dart        # Modello dati per una categoria

  screens/
    addBook.dart          # Schermata per aggiunta/modifica di un libro
    book_detail_page.dart # Dettaglio completo di un libro
    categories_page.dart  # Elenco di tutte le categorie disponibili
    category_detail_page.dart # Libri appartenenti a una determinata categoria
    favorites_page.dart   # Lista dei libri contrassegnati come preferiti
    Home_page.dart        # Scaffold principale condiviso tra le schermate
    home_screen.dart      # Schermata iniziale dell'app
    library_page.dart     # Libreria completa dei libri aggiunti manualmente
    myactivity_page.dart   # Statistiche di lettura (generi, voti, completati)
    popular_page.dart     # Lista estesa dei libri suggeriti (Popular now)
    search_page.dart      # Schermata di ricerca avanzata con filtri
    splashscreen.dart     # Schermata iniziale con animazione

  widgets/
    appBar_icon_button.dart # Widget riutilizzabile per pulsanti in AppBar
    book_card.dart          # Card compatta per rappresentare un libro
    book_grid_card.dart     # Card con layout a griglia
    bottom_nav_bar.dart     # Barra di navigazione inferiore

  main.dart               # Entry point dell'app (avvio app, DB, routing)
```

5. SCELTE PROGETTUALI

Lo sviluppo di *Library App* è stato orientato da scelte progettuali consapevoli, finalizzate a garantire un'elevata modularità del codice, una buona esperienza utente (UX) e una scalabilità funzionale dell'applicazione.

5.1 Framework e architettura

Flutter è stato scelto per la sua capacità di creare applicazioni multipiattaforma (iOS e Android) con una singola codebase. Offre prestazioni elevate grazie al rendering nativo e un'ampia flessibilità nella progettazione dell'interfaccia.

Architettura modulare: il codice è stato suddiviso in cartelle distinte per models, data, pages e widgets, seguendo il principio di separazione delle responsabilità, per una maggiore leggibilità e scalabilità.

5.2 Persistenza dei dati

SQLite, tramite il pacchetto sqflite, è stato integrato per la gestione locale dei dati in modalità offline. Sono stati implementati metodi CRUD (create, read, update, delete) per due entità principali: Book e Category.

5.3 Consistenza e integrità dei dati

Per garantire coerenza e affidabilità delle informazioni, sono stati introdotti **vincoli di unicità** sui nomi di libri e categorie. Non è possibile inserire due libri con lo stesso titolo né creare categorie con nomi duplicati (verifica case-insensitive).

5.4 Esperienza utente e responsività

L'interfaccia è progettata per essere completamente **responsive**, adattandosi dinamicamente a schermi piccoli, tablet e orientamenti orizzontali.

È stato introdotto un layout flessibile per le **card** (es. BookGridCard) con un limite massimo di larghezza, per evitare overflow e garantire una griglia armonica su qualsiasi dispositivo.

L'interazione avviene tramite elementi familiari come **icone**, **popup**, **menù a tendina**, e feedback visivi immediati (es. errori sotto i campi).

5.5 Gestione delle immagini

Le immagini dei libri non vengono memorizzate direttamente nel database, ma salvate come **file locali** nella memoria del dispositivo.

Nel campo imagePath del modello Book viene archiviato esclusivamente il percorso del file, per ottimizzare lo spazio occupato dal database e consentire all'utente il caricamento di immagini dalla galleria o fotocamera.

5.6 Navigazione tra schermate

La navigazione tra le schermate è stata gestita tramite `Navigator.push()` e `Navigator.pop()`, con passaggio diretto di oggetti (es. `Book`) tra le view.

Questa scelta ha permesso di aprire schermate dettagliate (es. `BookDetailPage`) con i dati contestuali, riutilizzare `AddBookPage` sia per la creazione che per la modifica di un libro e garantire coerenza tra gli stati condivisi (lettura, categoria, preferiti) in tutte le schermate coinvolte.

5.7 Versionamento e collaborazione

Il codice sorgente è stato gestito tramite **Git** e **GitHub**, con l'utilizzo di branch per funzionalità e merge controllati. Questo ha permesso una collaborazione efficace e tracciabilità completa dello sviluppo.

5.8 Gestione dei pacchetti

Nel file `pubspec.yaml` sono stati dichiarati i pacchetti fondamentali per il funzionamento dell'app:

- `flutter`: SDK principale che fornisce widget, gestione dello stato, layout responsive e accesso alle risorse locali.
- `cupertino_icons`: icone in stile iOS per una UI coerente su tutte le piattaforme.
- `image_picker`: consente la selezione di immagini dalla galleria o fotocamera.
- `intl`: utilizzato per la formattazione e localizzazione delle date (es. selezione di mese/anno in fase di completamento di un libro).

6. DESCRIZIONE DETTAGLIATA DELL'APPLICAZIONE

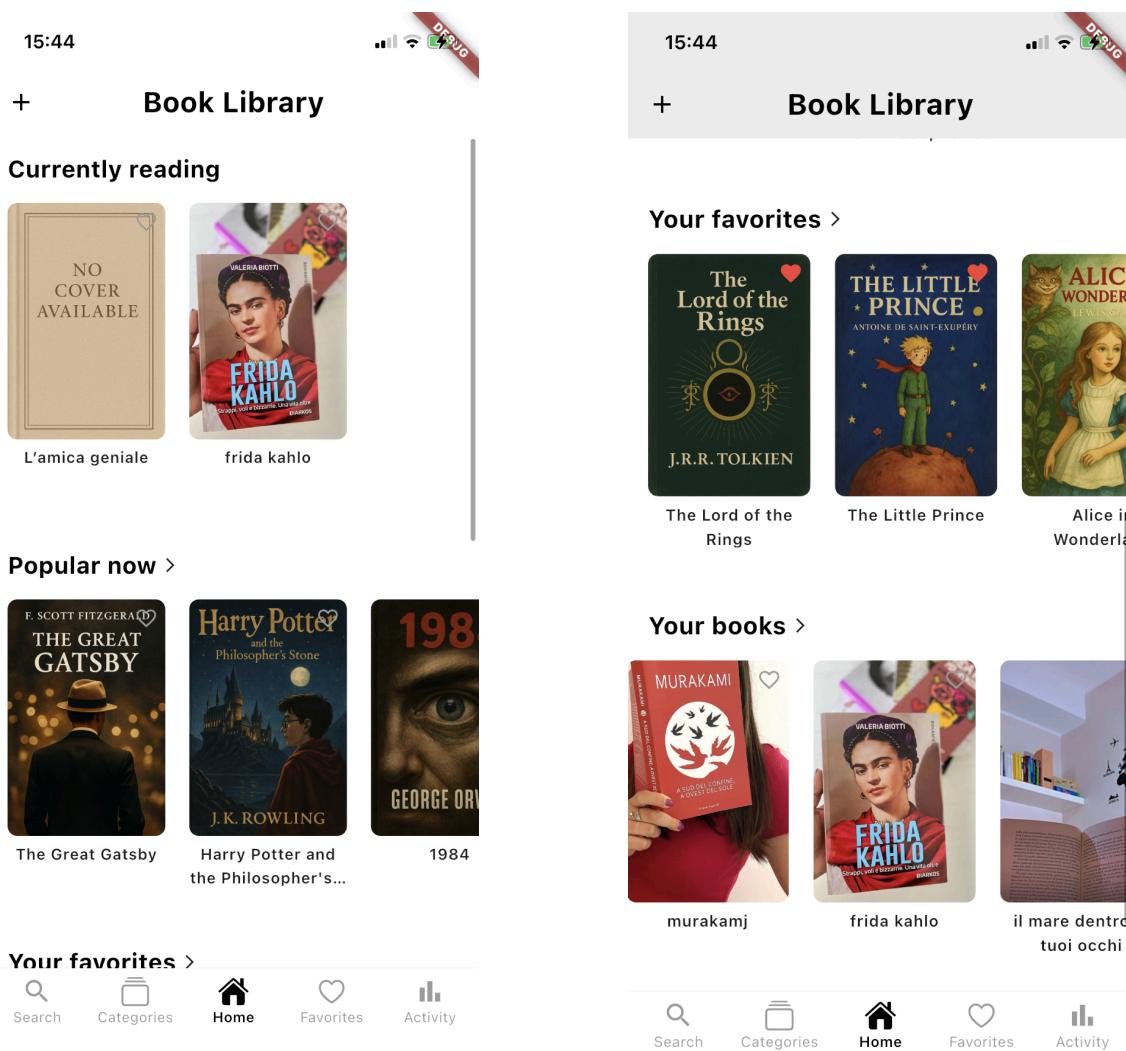
Per rappresentare in modo esaustivo la struttura funzionale dell'interfaccia, sono stati acquisiti screenshot delle principali schermate dell'applicazione. Di seguito viene fornita una panoramica dettagliata dei layout, dei componenti UI implementati e delle logiche di interazione utente associate a ciascuna sezione.

6.1 Schermata principale (Home)

La schermata iniziale dell'app rappresenta il punto di accesso principale alle funzionalità chiave. La UI è articolata in una serie di sezioni scrollabili verticalmente:

- Un Floating Action Button "+" in alto a sinistra, che consente di accedere rapidamente alla schermata di aggiunta di un nuovo libro;
 - La sezione "Currently Reading", che visualizza i libri il cui stato è impostato su *Reading*;
 - "Popular Now", contenente una lista orizzontale di libri suggeriti e predefiniti, non modificabili ma rimovibili;
 - "Your Favorites", con i libri contrassegnati come preferiti;
- "Your Books", che mostra gli ultimi libri aggiunti manualmente dall'utente.

Ogni sezione, eccetto la prima, consente l'accesso a una schermata estesa mediante un pulsante di navigazione contestuale (">"). La *bottom navigation bar*, persistente su ogni schermata, garantisce accesso diretto alle sezioni *Search*, *Categories*, *Favorites*, *Activity* e *Home*. La sezione "Currently Reading" non è estendibile in un'altra schermata perché si assume che l'utente non legga molti libri contemporaneamente



6.2 Schermata aggiunta/modifica libro

La schermata *AddBookPage* consente all'utente di inserire un nuovo libro o modificarne uno esistente. I campi gestiti sono:

- Titolo
- Autore
- Trama
- Genere (con selezione a tendina)
- Stato di lettura (*To Read*, *Reading*, *Completed*) con eventuale selezione di mese/anno se completato
- Immagine di copertina, caricabile da file
- Categoria, selezionabile o creabile al volo tramite popup

L'accesso in modalità modifica è abilitato esclusivamente per i libri inseriti manualmente. I libri precaricati nella sezione *Popular Now* non sono modificabili.

15:53 DEBUG

Add Book

Title _____

Author _____

Plot _____

Genre _____
Fantasy

State _____
To Read


Tap to add cover image

+ Add book

16:00 DEBUG

Edit Book

Title _____
il mare dentro ai tuoi occhi

Author _____
elia palesa

Plot _____

Genre _____
Drama

State _____
Reading



Save changes

6.3 Schermata dettaglio libro

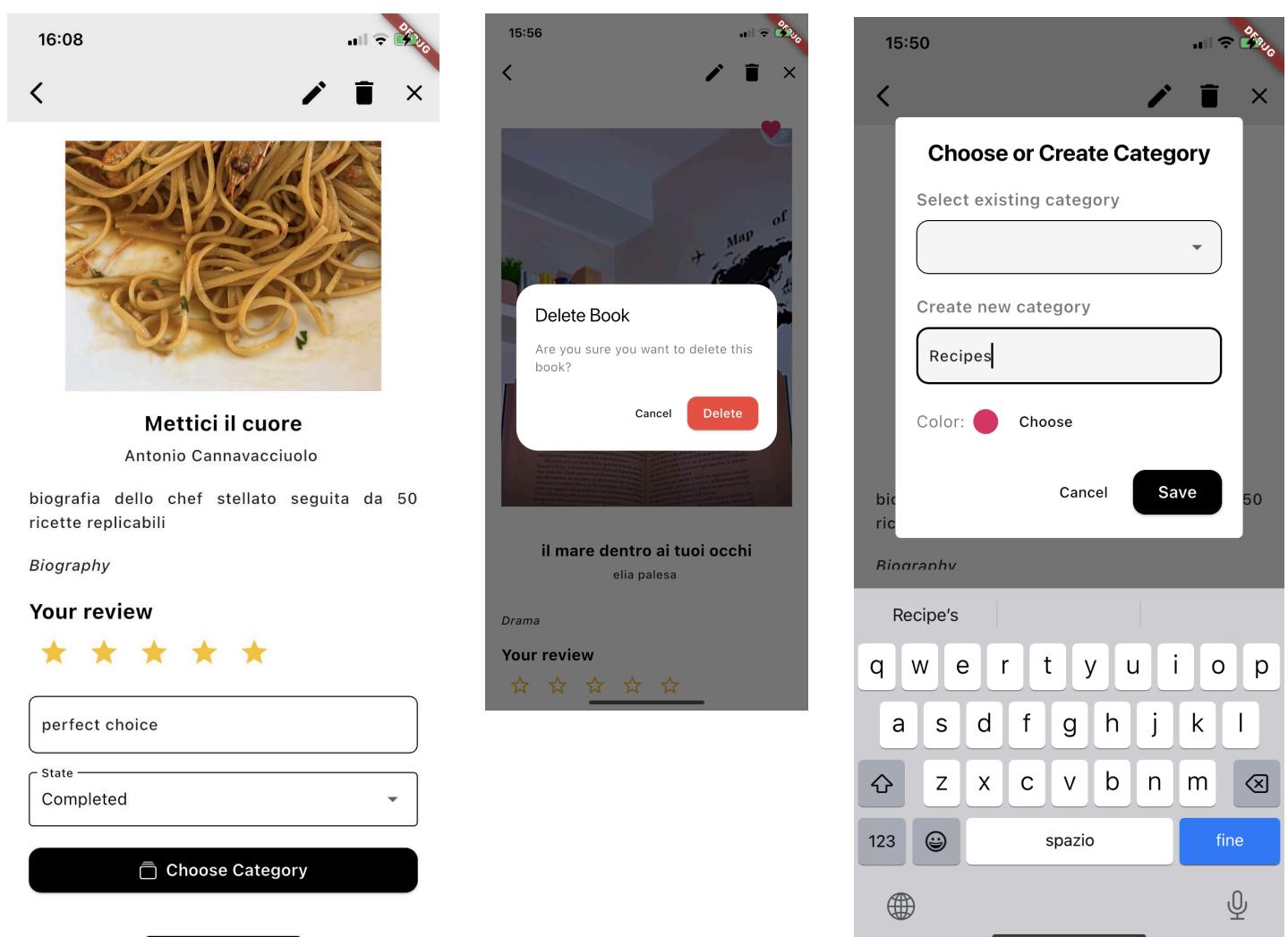
La schermata *BookDetailPage* visualizza nel dettaglio tutte le proprietà di un libro selezionato:

- Copertina, titolo, autore, trama
- Stato di lettura e valutazione in stelle
- Commento opzionale e categoria assegnata

Dalla schermata è possibile:

- Modificare il libro (tramite icona a forma di matita, visibile solo per i libri utente)
- Assegnare una categoria tramite popup dedicato
- Contrassegnare o rimuovere il libro dai preferiti
- Eliminare il libro in modo definitivo

L'interfaccia si adatta dinamicamente allo stato del libro (ad es. un libro "Completed" mostra anche la data di completamento).



6.4 Schermata categoria

La schermata *CategoriesPage* mostra un elenco visuale delle categorie personalizzate, ognuna rappresentata da:

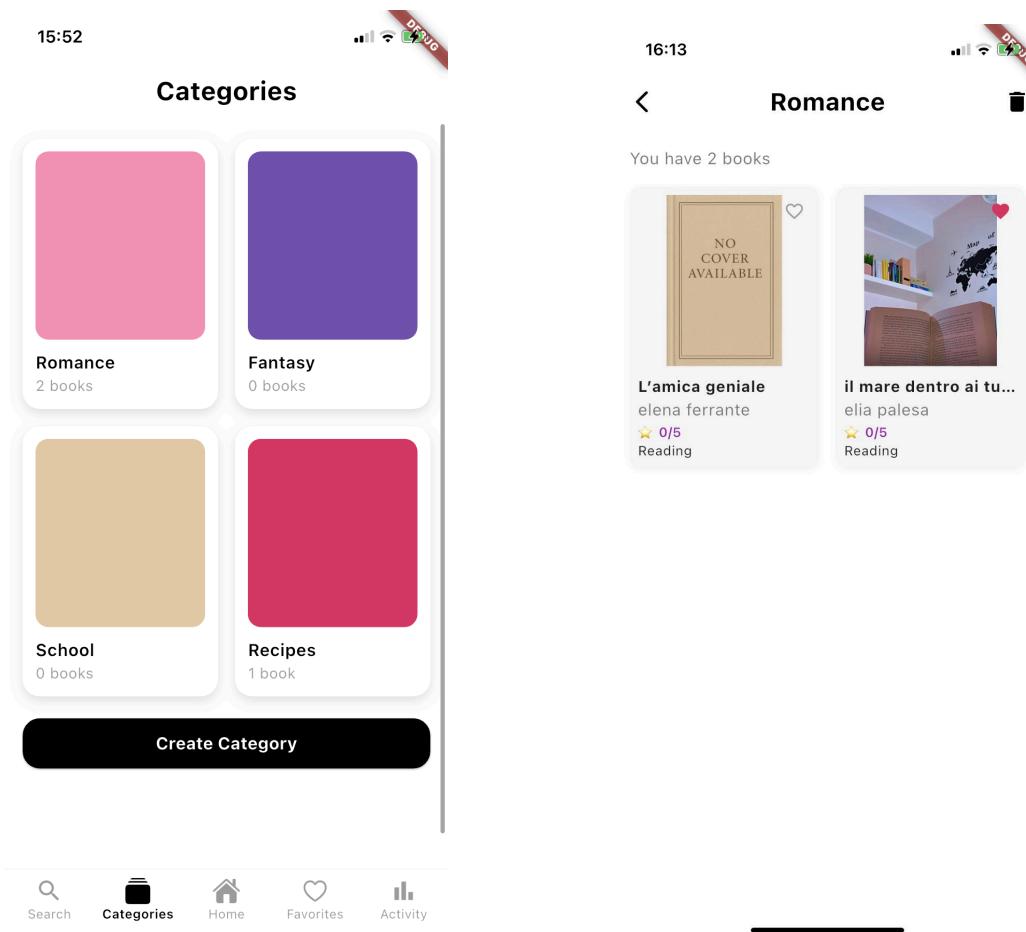
- Nome univoco
- Colore distintivo selezionato tramite color picker
- Conteggio dei libri associati

La creazione avviene tramite popup modale che gestisce sia l'inserimento del nome che la selezione del colore, con controllo integrato sulla duplicazione. Il tap su una categoria conduce alla relativa schermata di dettaglio.

6.5 Schermata dettaglio categoria

Espone l'elenco filtrato dei libri appartenenti a una specifica categoria. I libri sono visualizzati tramite card disposte a griglia, con azione di tap per accedere alla schermata di dettaglio.

La AppBar include un'icona *delete* che consente la rimozione della categoria: i libri associati non vengono eliminati ma semplicemente dissociati.

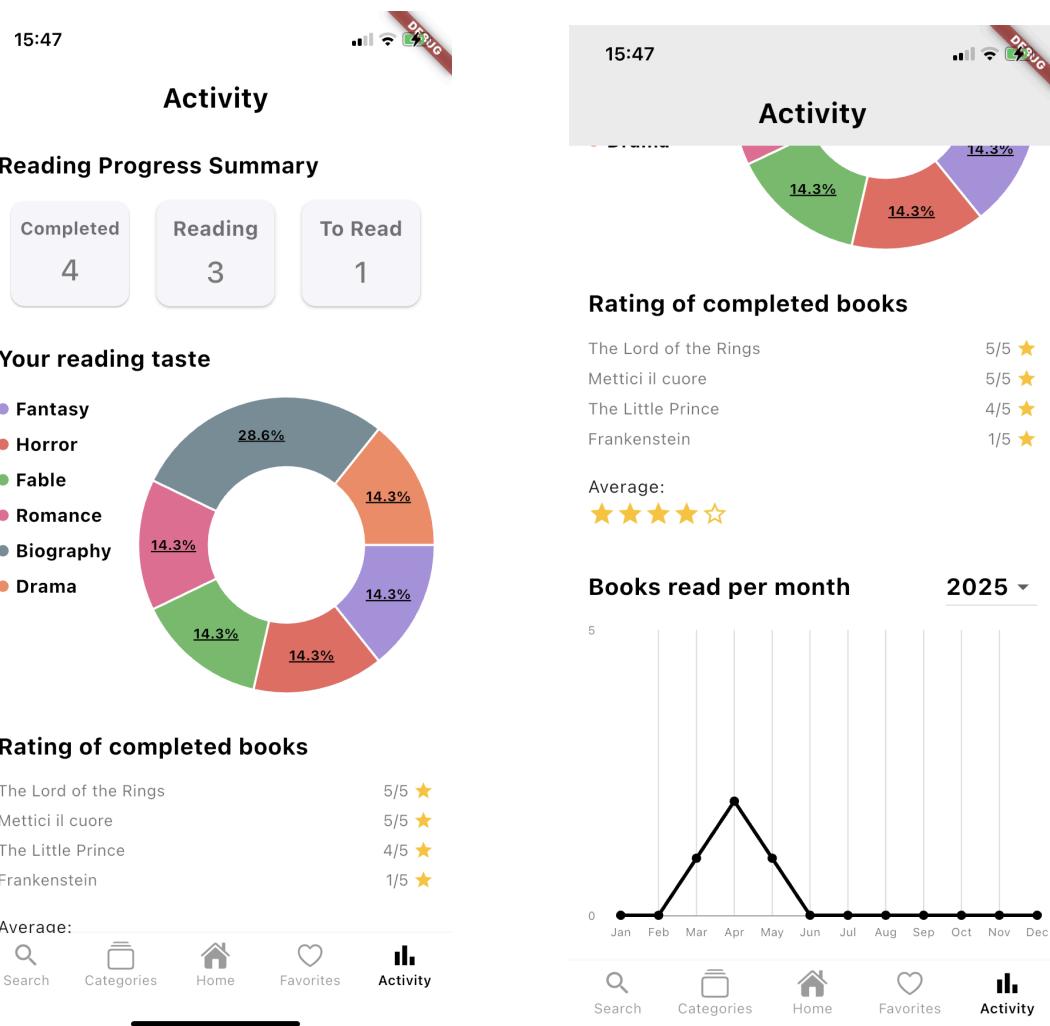


6.6 Schermata analisi (Activity)

La schermata *MyActivityPage* fornisce una panoramica statistica delle abitudini di lettura dell'utente. I dati visualizzati includono:

- Reading Progress Summary: numero totale di libri *Completed*, *Reading* e *To Read*
- Your Reading Taste: grafico a torta per la distribuzione per genere
- Rating of Completed Books: valutazione media in stelle
- Reading Timeline (*opzionale*): andamento mensile/annuale dei libri completati

Tutti i dati sono visualizzati tramite widget responsive, adattabili a diverse dimensioni di schermo.



6.7 Schermata ricerca

La schermata *SearchPage* consente la ricerca avanzata dei libri attraverso:

- Campo testuale con ricerca per titolo, autore o entrambi
- Dropdown per filtri multipli: ordine alfabetico, stato di lettura, genere e valutazione
- Sezione Recommended, che mostra risultati dinamici in base ai criteri selezionati

La voce *Results* appare solo in presenza di match attivi, altrimenti viene mostrata la sezione *Previous Searches*, che memorizza gli ultimi libri aperti durante le ricerche.

