**Virtual library app**

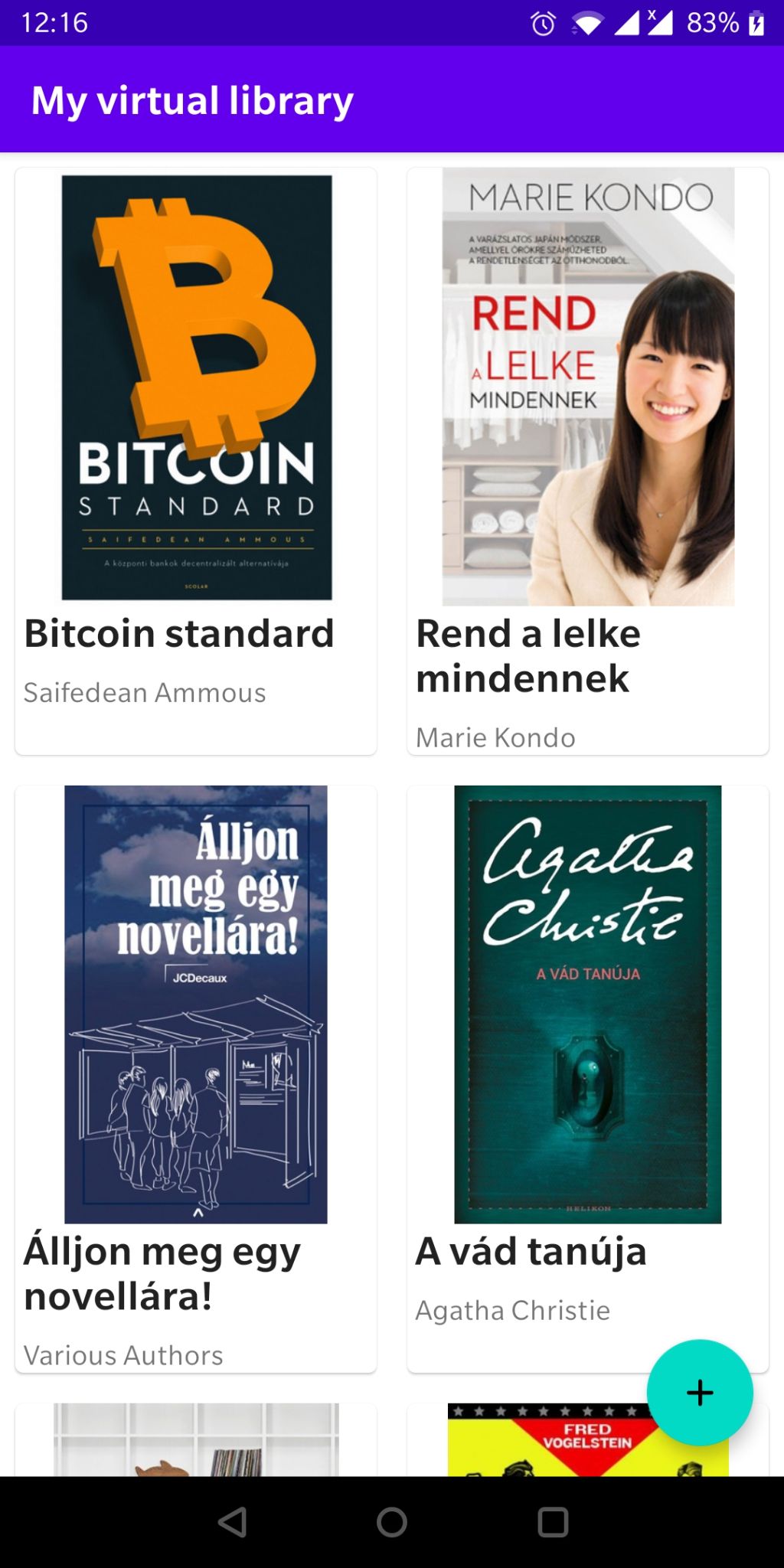
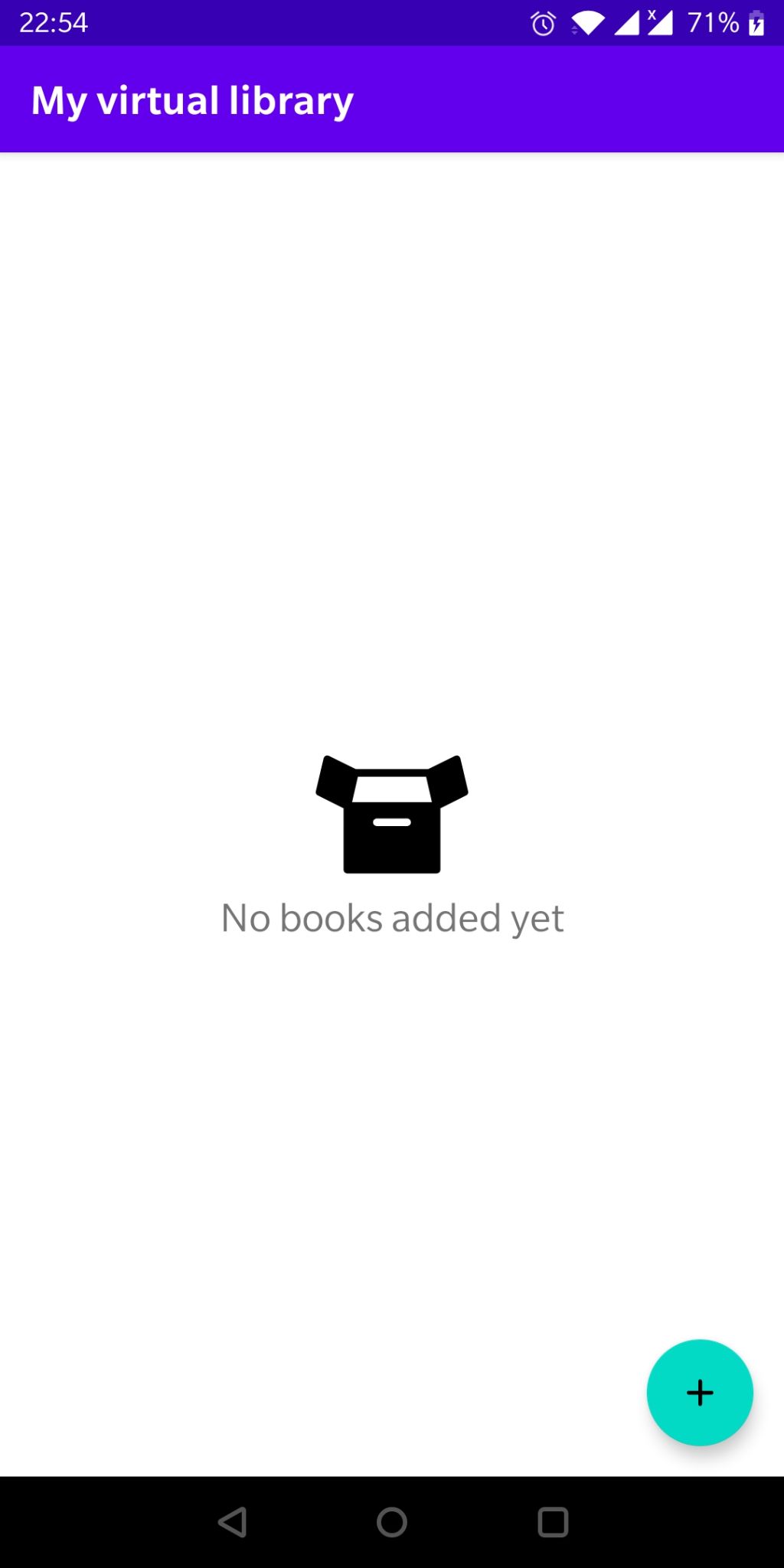
**Project goal:**

* Create an application which allows users to add books to a list by scanning its barcode or enter its details manually, and stores that list in a database.

**Features:**

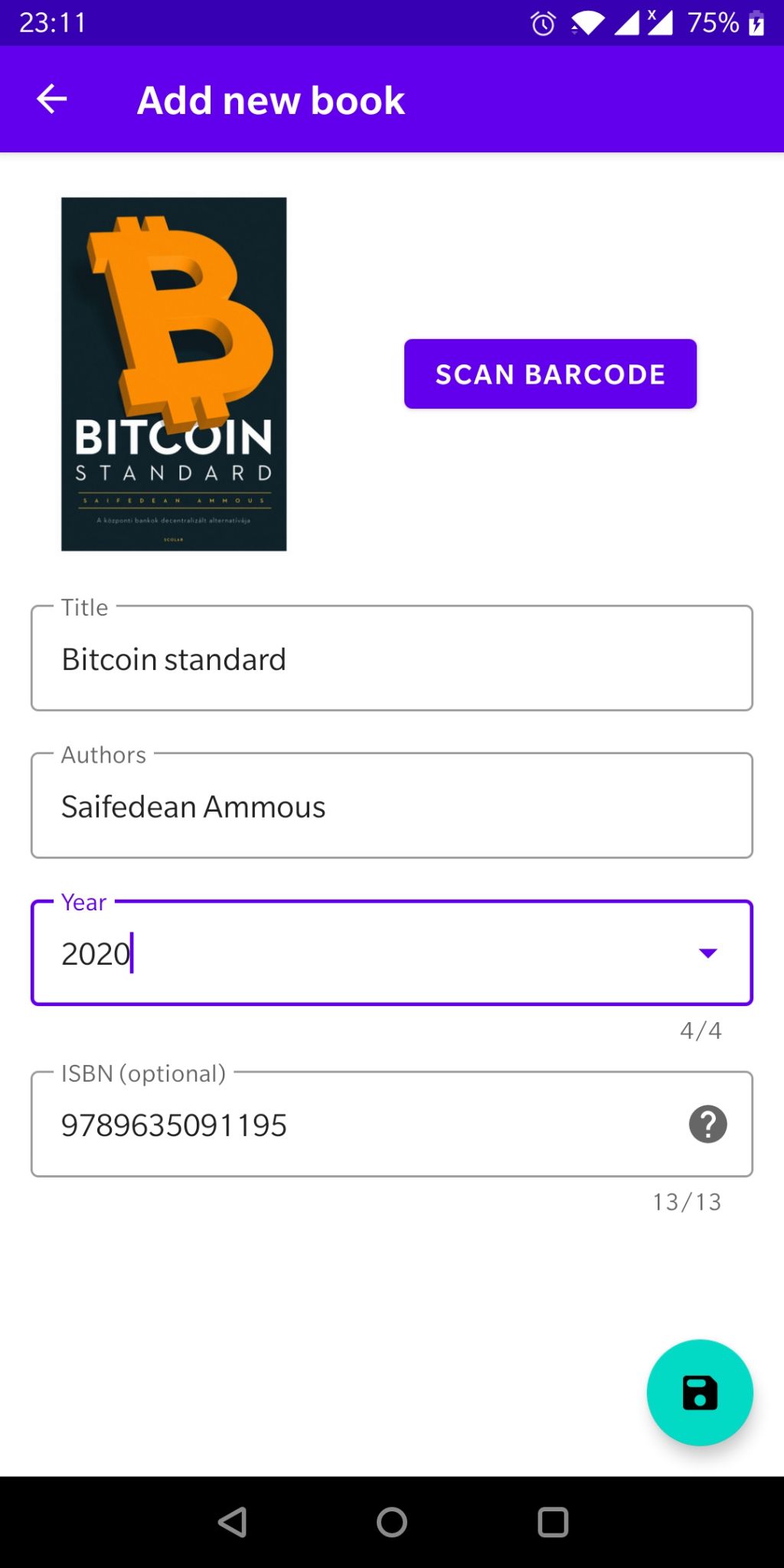
* Single activity app using Navigation Controller for fragment navigation.
* App stores data in a Room database

**ListFragment:**



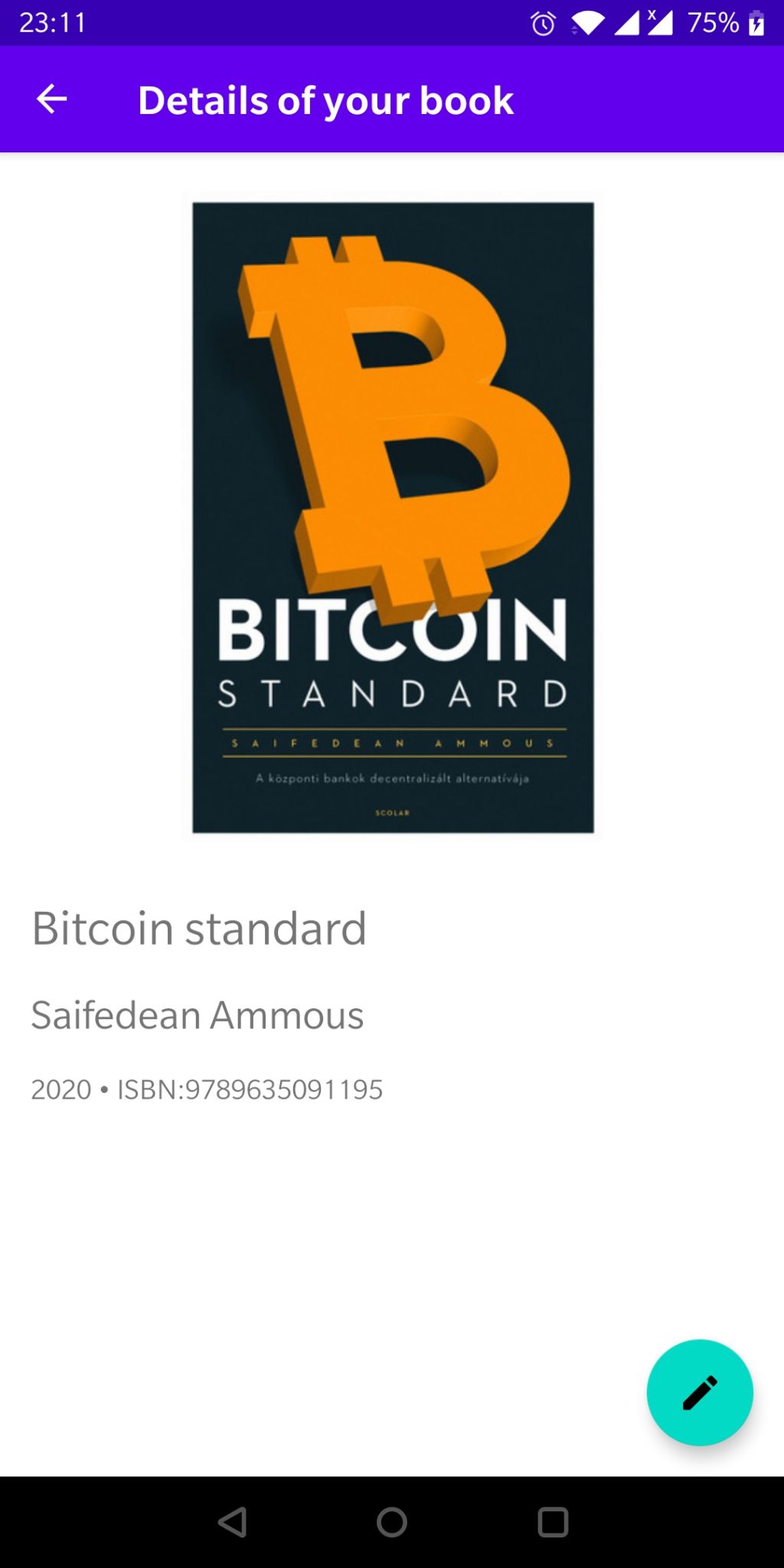
* Shows the list of stored books using RecyclerView and GridLayoutManager
* When there aren’t any stored books, the screen show “No data” indicator (left photo)
* If you click on the box icon, it will do an animation using MotionLayout, defined in MotionScene using Transition nodes and ConstraintSet blocks
* When user clicks on a selected book, the app navigate to DetailFragment to show stored information

**AddFragment:**

****

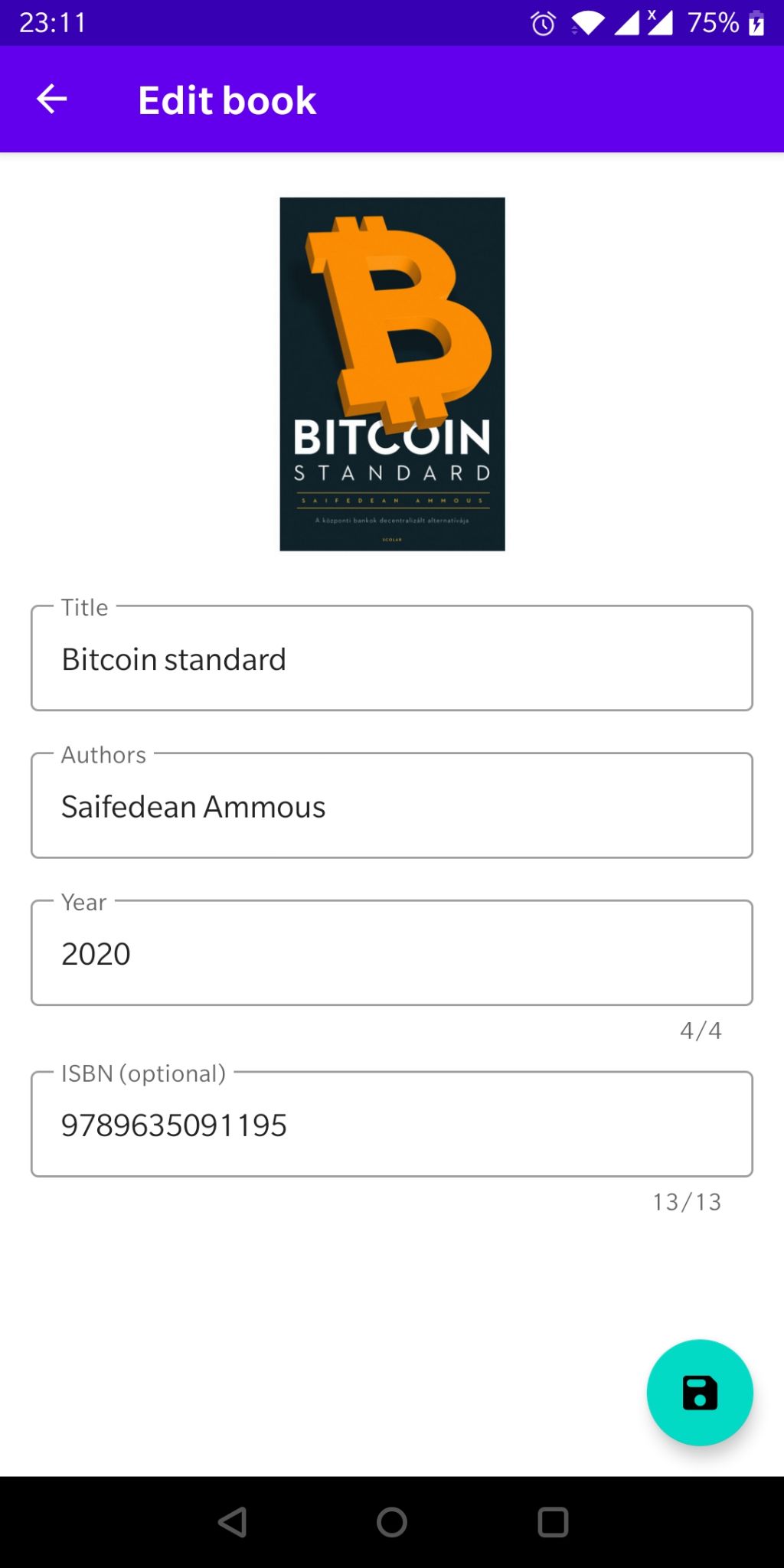
* User can add new books by typing its details manually
* User can add their book’s details by scanning its barcode using ***CameraActivity***
  + After getting the raw value, it uses Retrofit for REST API request informations about scanned ISBN value
  + Retrived data held in local models using Moshi
* If we got proper info about the book, it will fill the EditTexts + fills the dropdown list of release years
* Also it will try to load the cover photo using Glide, if can’t it shows a broken image icon
* If user clicks on the question mark icon, it will show a dialog about ISBN
* When user clicks on Save FAB, the app checks that title, author and year are all given (ISBN is optional due to antique books)

**DetailFragment:**

****

* Loads book data from local Room database using its ID (passed with safe-args)
* Showing selected book’s details: title, author, year of publish and ISBN if we have
* If user clicks on Edit FAB, it will navigate to EditFragment to update it

**EditFragment:**

****

* Change part of a book and update its information
* User can’t change the cover photo
* If the user clicks on Save FAB the app
  + Checks for required fields
  + Updates in the database
  + Navigate back to DetailFragment

**CameraActivity:**

* Asking user for camera permission to be able to scan barcode
  + Permission requested at runtime
* Plain camera view only which is being analyzed using
  + CameraX for camera footage and to analyze image
  + MLKit Vision - Barcode scanning to decode the value of the scanned barcode