Predict Spring Mobile Product Catalog App

Requirements
1. Use an iPad/iPhone and Swift language ✓
2. The iOS app takes the file name as input. ✓
3. The app streams the file into memory or a local file ✓
4. Process the file for records by parsing the rows and columns ✓
5. Insert records into SQLite database ✓
6. Provide stats on progress while processing the file. ✓
7. Provide a single screen to search products from the db ✓
8. Display the products in a table view. (You can display 10 or 20 at a time and support scroll) ✓

App Design

Classes

1. MVVM design pattern is followed.
2. Product - ProductsListViewController - ProductViewModel
3. ProductsListViewController is the viewcontroller, Product is the Model file, ProductsViewModel is the corresponding ViewModel.
4. DownloadManager - Includes methods for the download operations
5. DatabaseManager - Includes methods for all the db related operations
6. All the calls to DatabaseManager/DownloadManager happens only in the ProductsViewModel.

App functionality

1. UI implemented using UIKit, No storyboard used as the UI creation and autolayout constraints are written programmatically.
2. Features
    a. Download
        i. The given file is downloaded from the GoogleDrive asynchronously and saved locally using URLSession's downloadtask API, a progress bar showing download status is updated on the main thread.
    b. Insert
        i. The file has 1000000 records. After the file has been downloaded, a process is initiated to insert the 1000000 records from the CSV file to the Sqlite database.
        ii. Number of records processed/ stats is shown in the UI
        iii. The background process is executed in a global queue with Qos - .userinitiated and the UI is updated on the main thread

    c. Display
        i. After all the records are inserted into the SQLite database, they are fetched from the database for display. 20 records are displayed initially and as the user scrolls more records are loaded (in the increments of 20)
    d. Search
        i. After the insert operation is completed, the Search Bar is enabled for data lookup by productId. As the user types in the search bar, the records matching the productId are fetched and are displayed. The records are updated as search text changes.
3. Dark mode/light mode compatible
4. For reactive/declarative programming SwiftUI/Combine could be used.
5. Testing - Unit Tests are written to test core functionality - Download and Database operations using XCTest framework

Note: Please wait for the background process to kick in after download is complete.
Delete the app again and reinstall to see the entire flow again.

Video of the app
https://drive.google.com/file/d/1C6XG6-6SNDugFQUFueIiplYBr5rGPweZ/view?usp=sharing