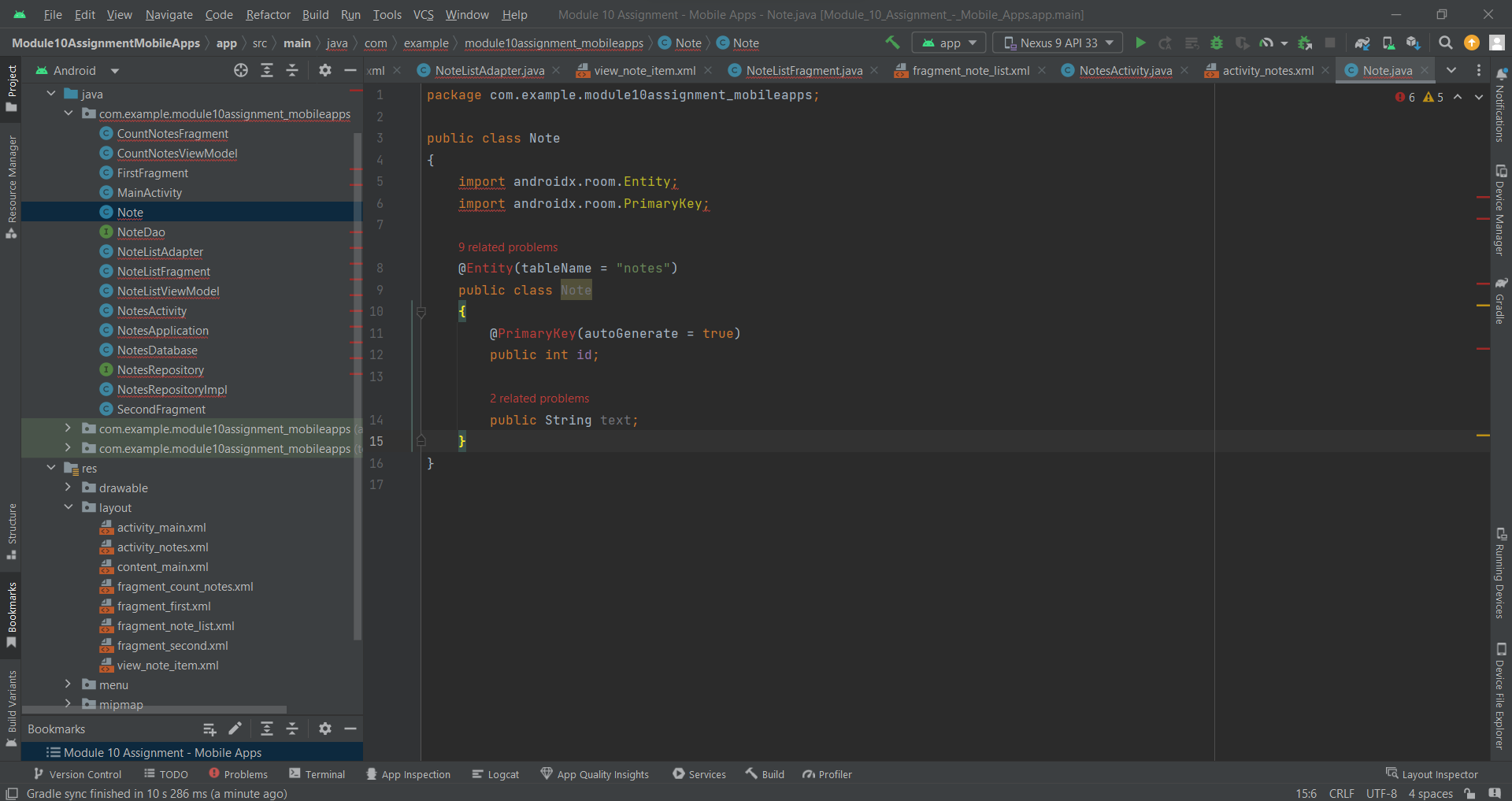
1. Note.java (Entity Class):

This file represents the Note entity in the Room database.

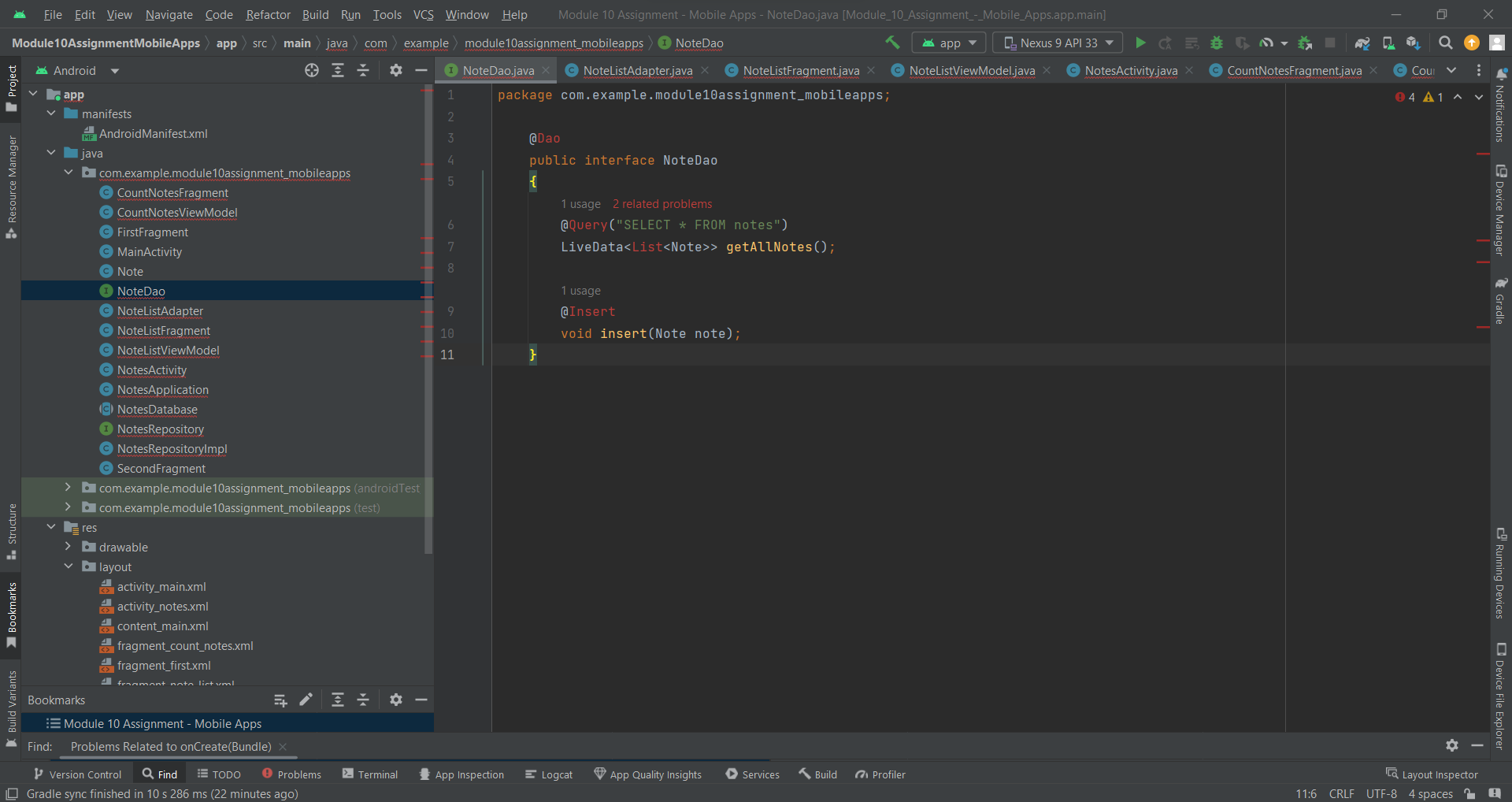
It contains fields such as id and text to store the note information.



1. NotesDao.java (Data Access Object):

This interface defines the methods to interact with the Room database.

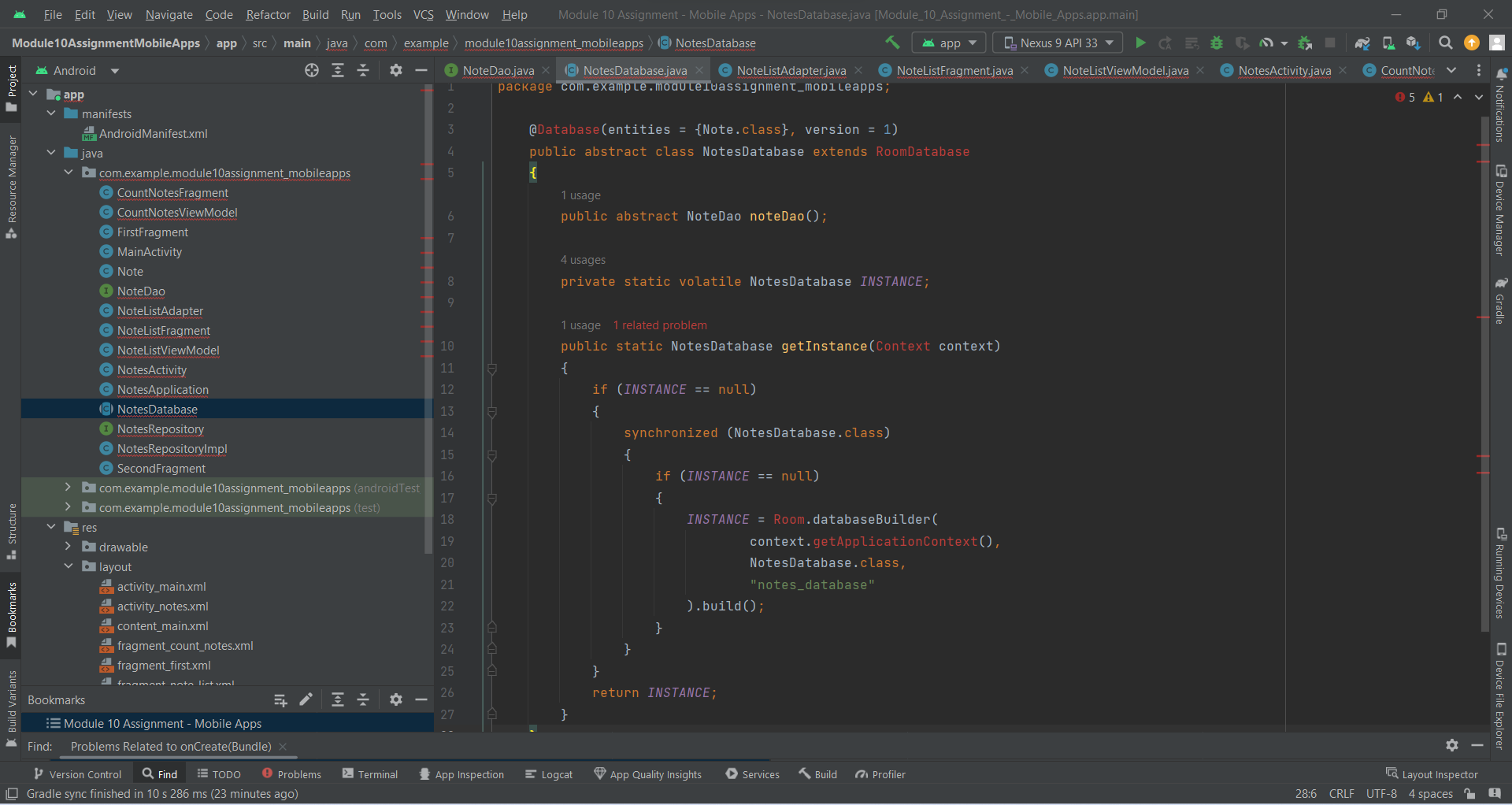
It includes methods such as inserting a new note and retrieving all notes.



1. NotesDatabase.java (Room Database):

This class represents the Room database.

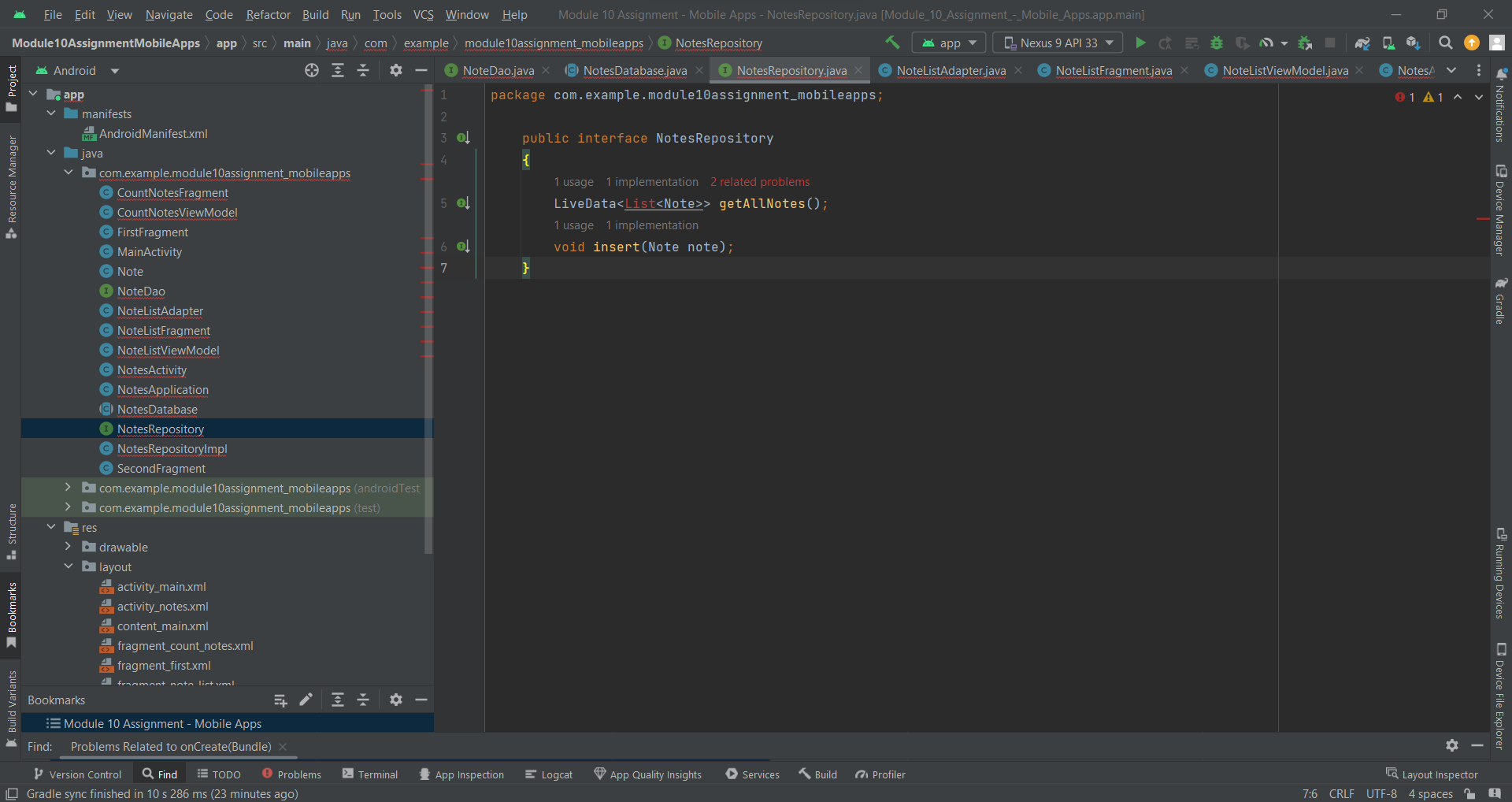
It defines the entities, sets up the database configuration, and provides the access methods through the DAO.



1. NotesRepository.java (Repository Interface):

This interface defines the repository methods for managing shopping notes.

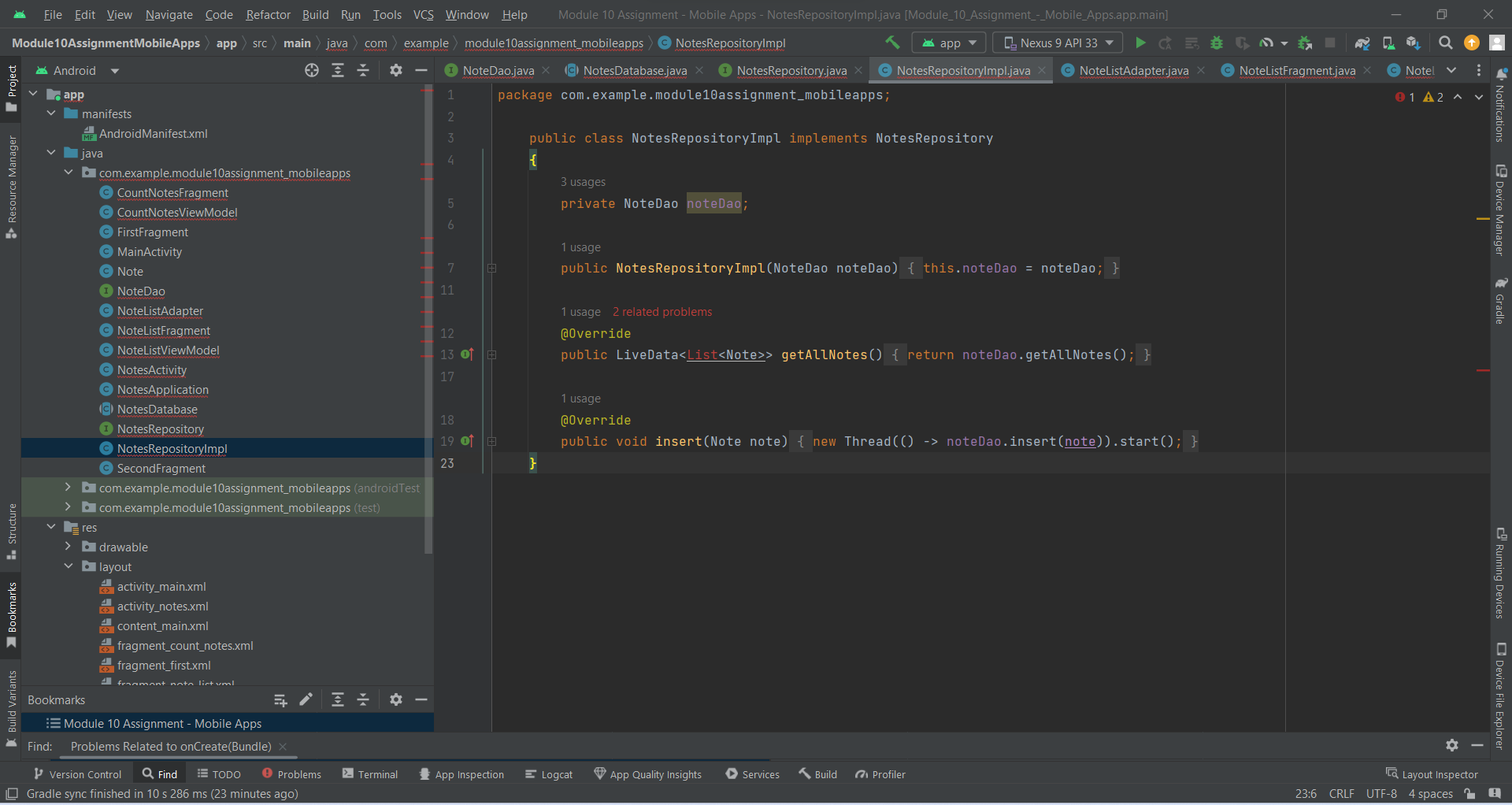
It includes methods such as retrieving all notes and inserting a new note.



1. NotesRepositoryImpl.java (Repository Implementation):

This class implements the NotesRepository interface.

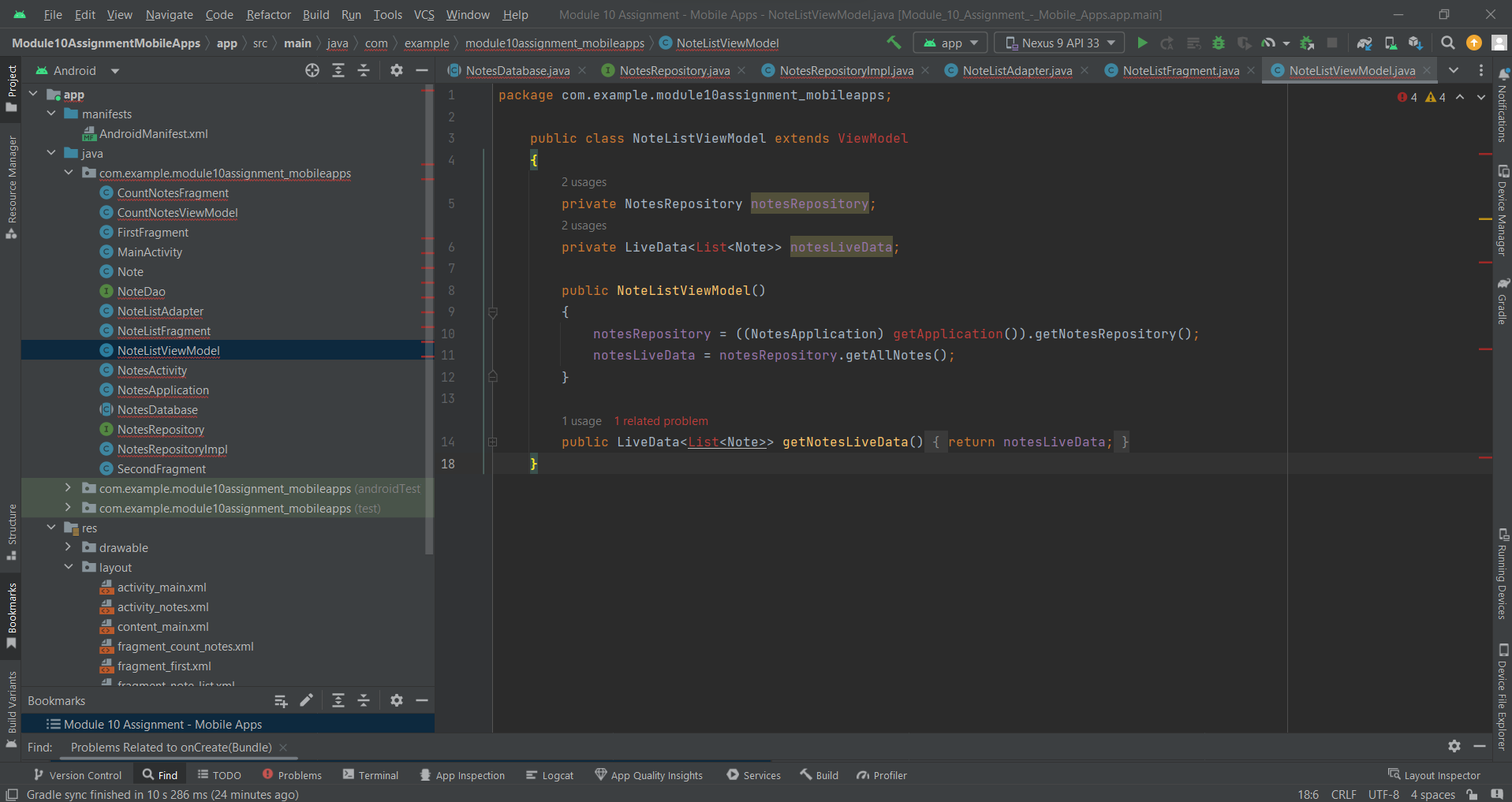
It handles the data operations and business logic related to shopping notes, such as inserting a note into the database.



1. NoteListViewModel.java:

This ViewModel class provides the data for the NoteListFragment.

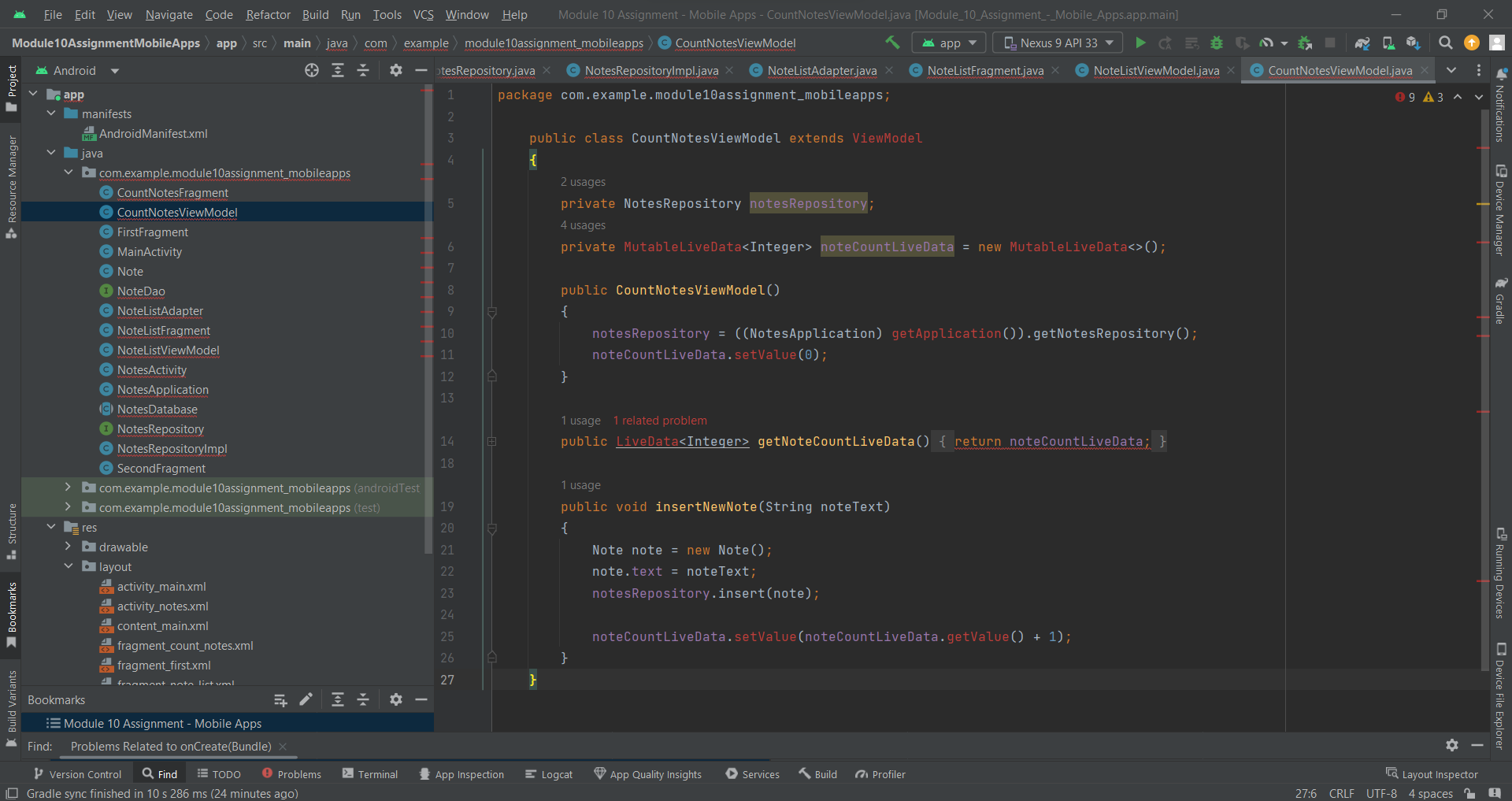
It retrieves the list of notes from the repository and exposes it as LiveData for observation.



1. CountNotesViewModel.java:

This ViewModel class provides the data for the CountNotesFragment.

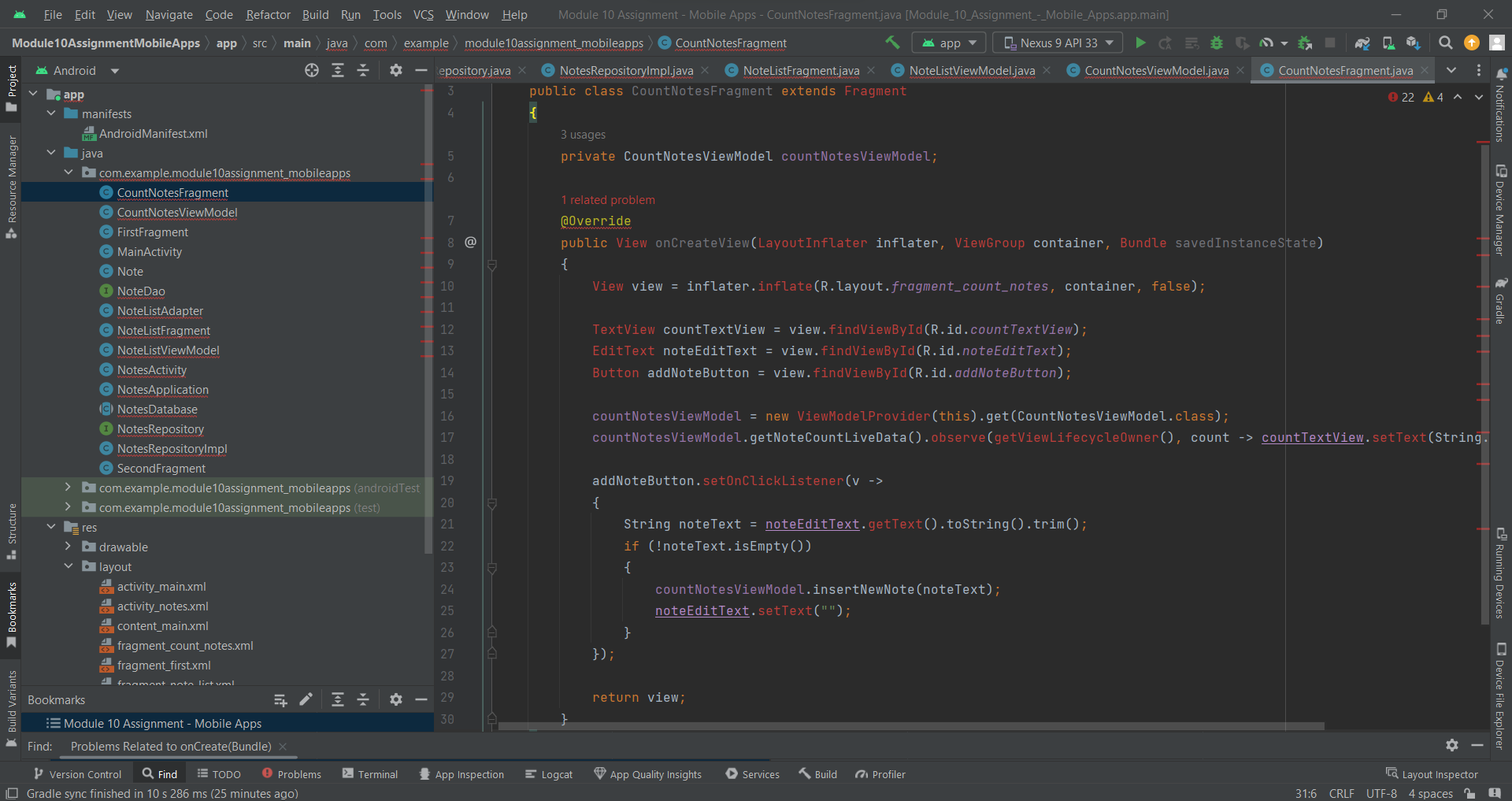
It retrieves the total number of notes from the repository as LiveData and handles inserting new notes.



1. CountNotesFragment.java:

This fragment displays the total number of notes and allows users to add new notes.

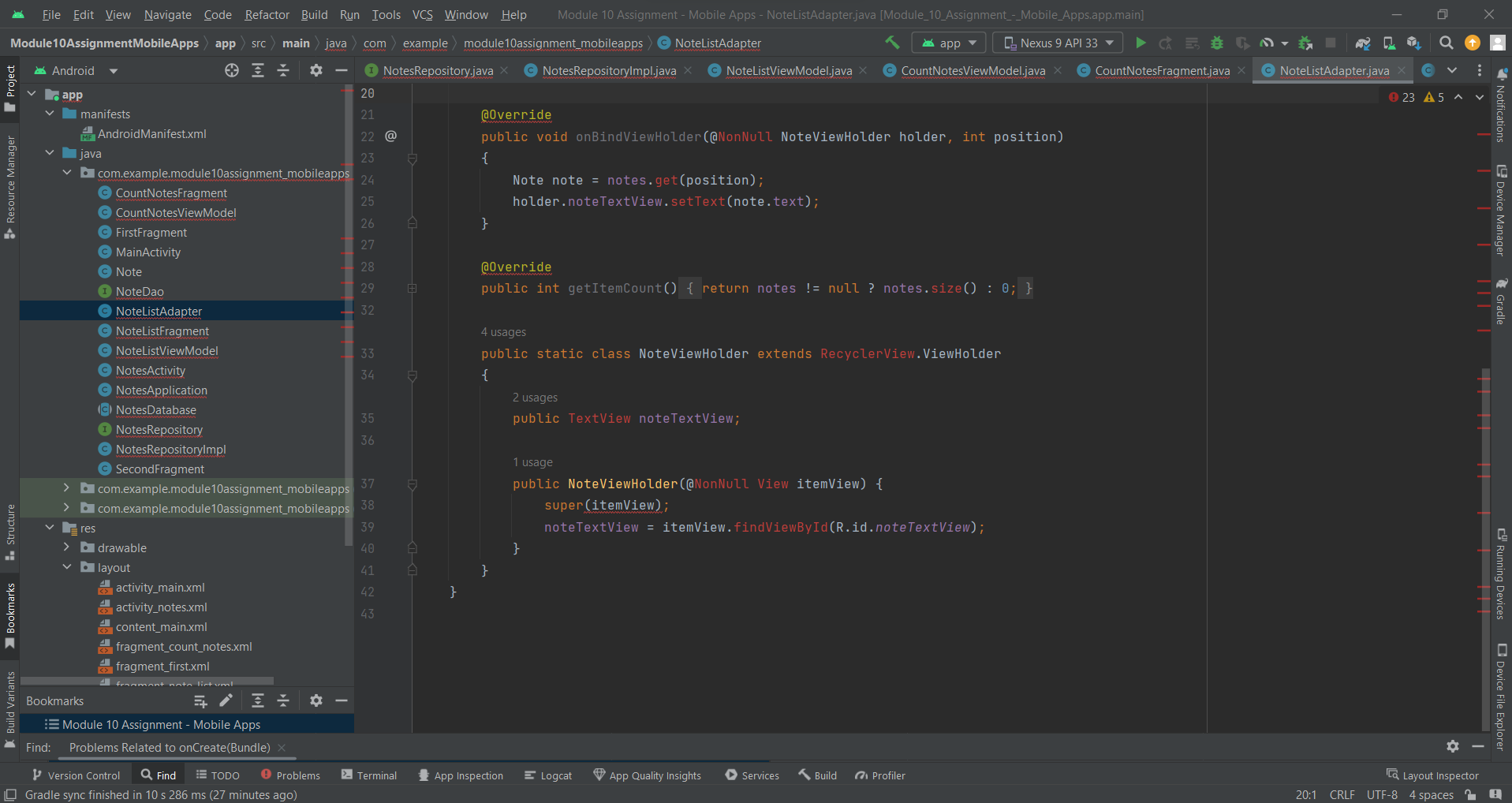
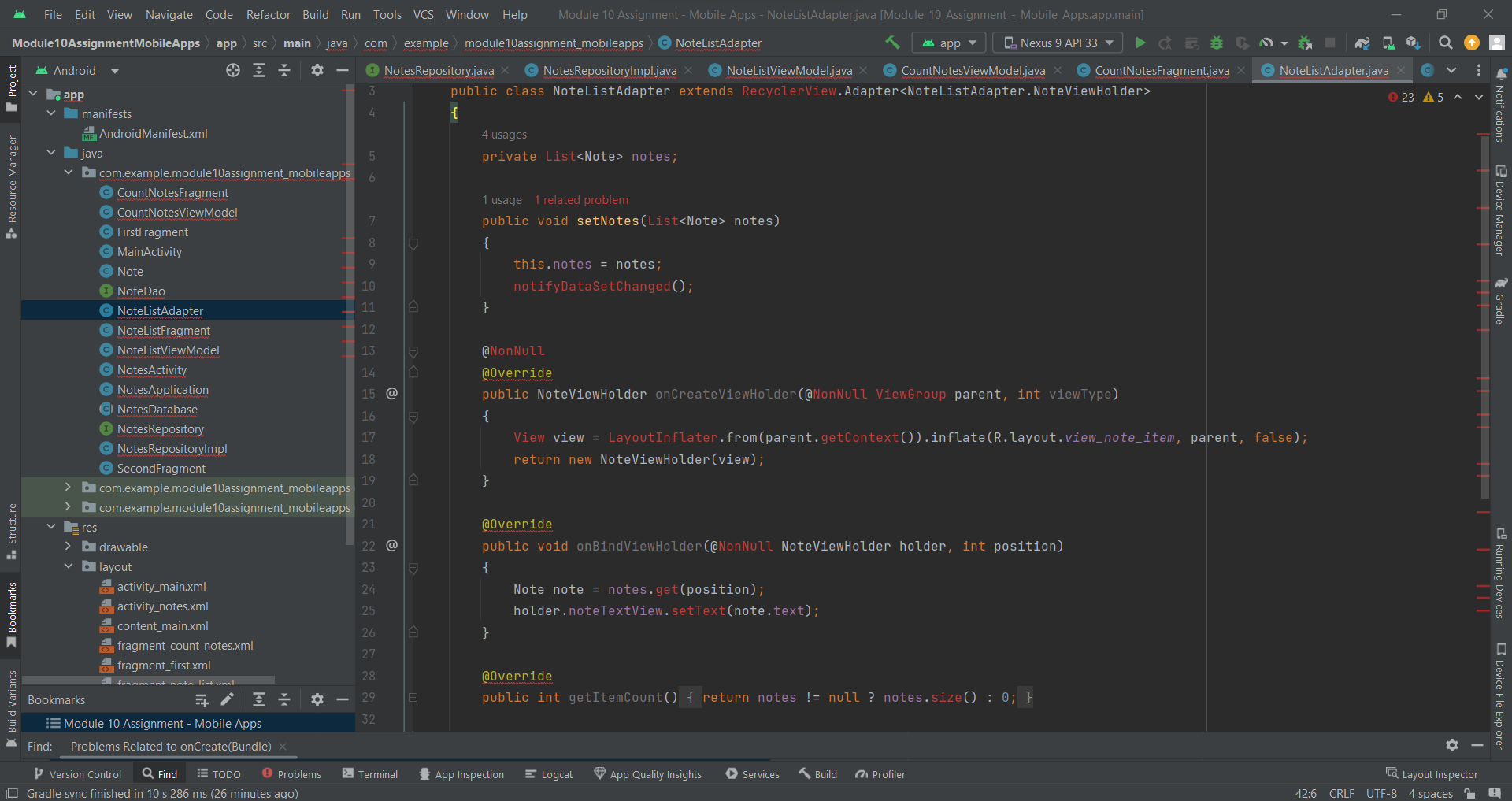
It observes the LiveData from CountNotesViewModel to update the UI.



1. NoteListAdapter.java:

This adapter class manages the data for the RecyclerView in NoteListFragment.

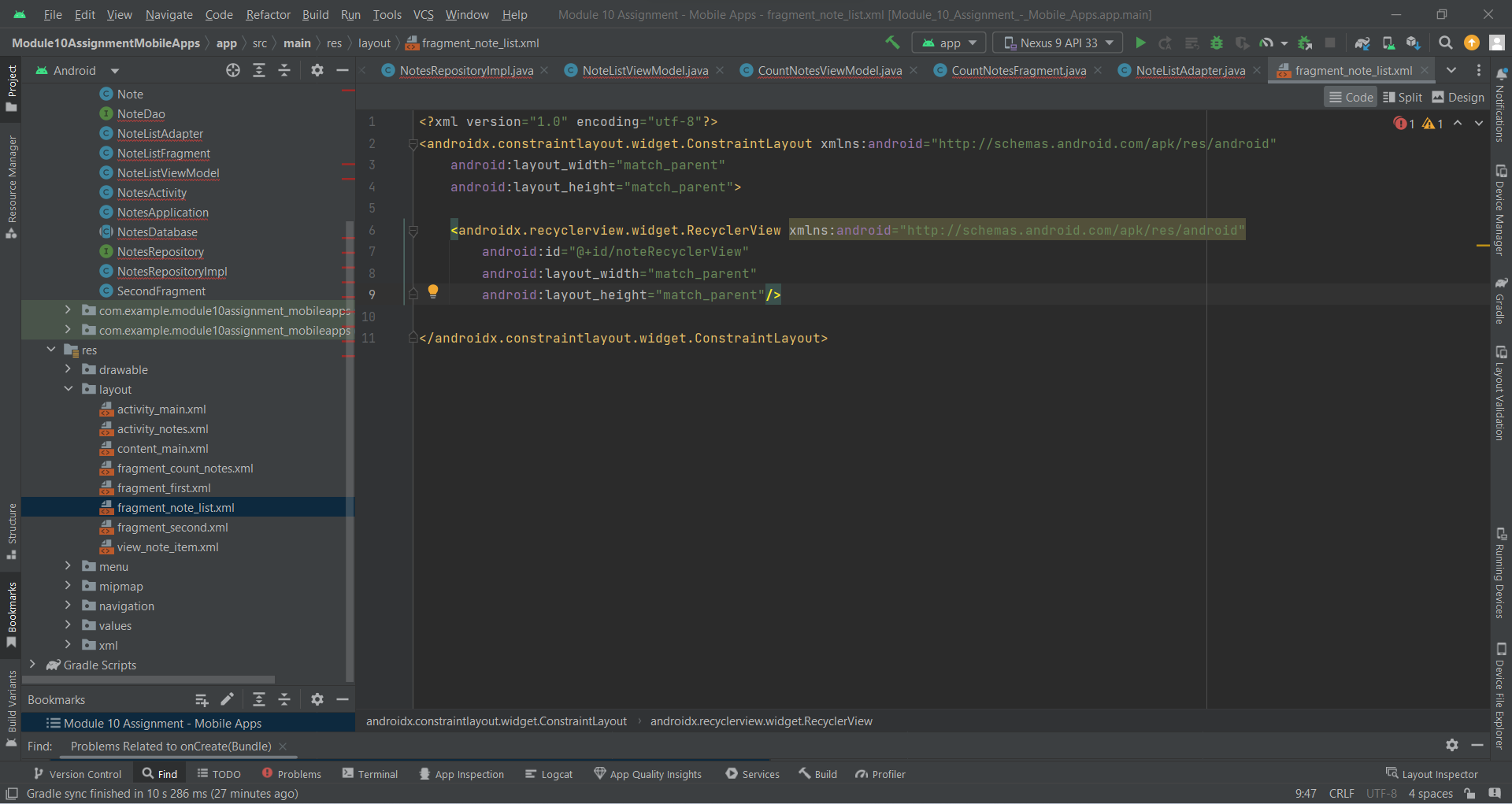
It binds the note data to the item view and handles item click events.



1. fragment\_note\_list.xml:

This layout file defines the UI for the NoteListFragment.

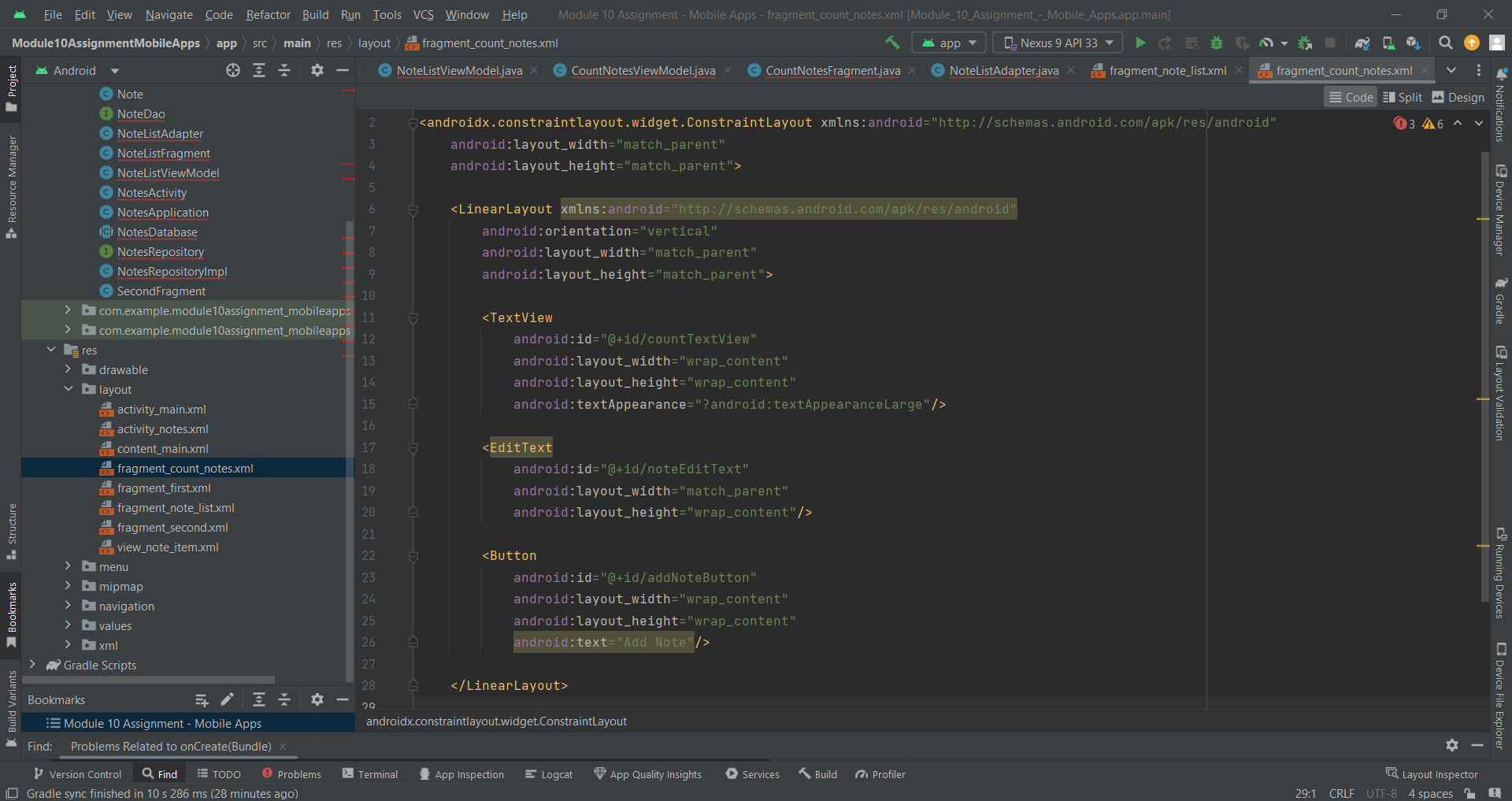
It contains a RecyclerView to display the list of notes using NoteListAdapter.



1. fragment\_count\_notes.xml:

This layout file defines the UI for the CountNotesFragment.

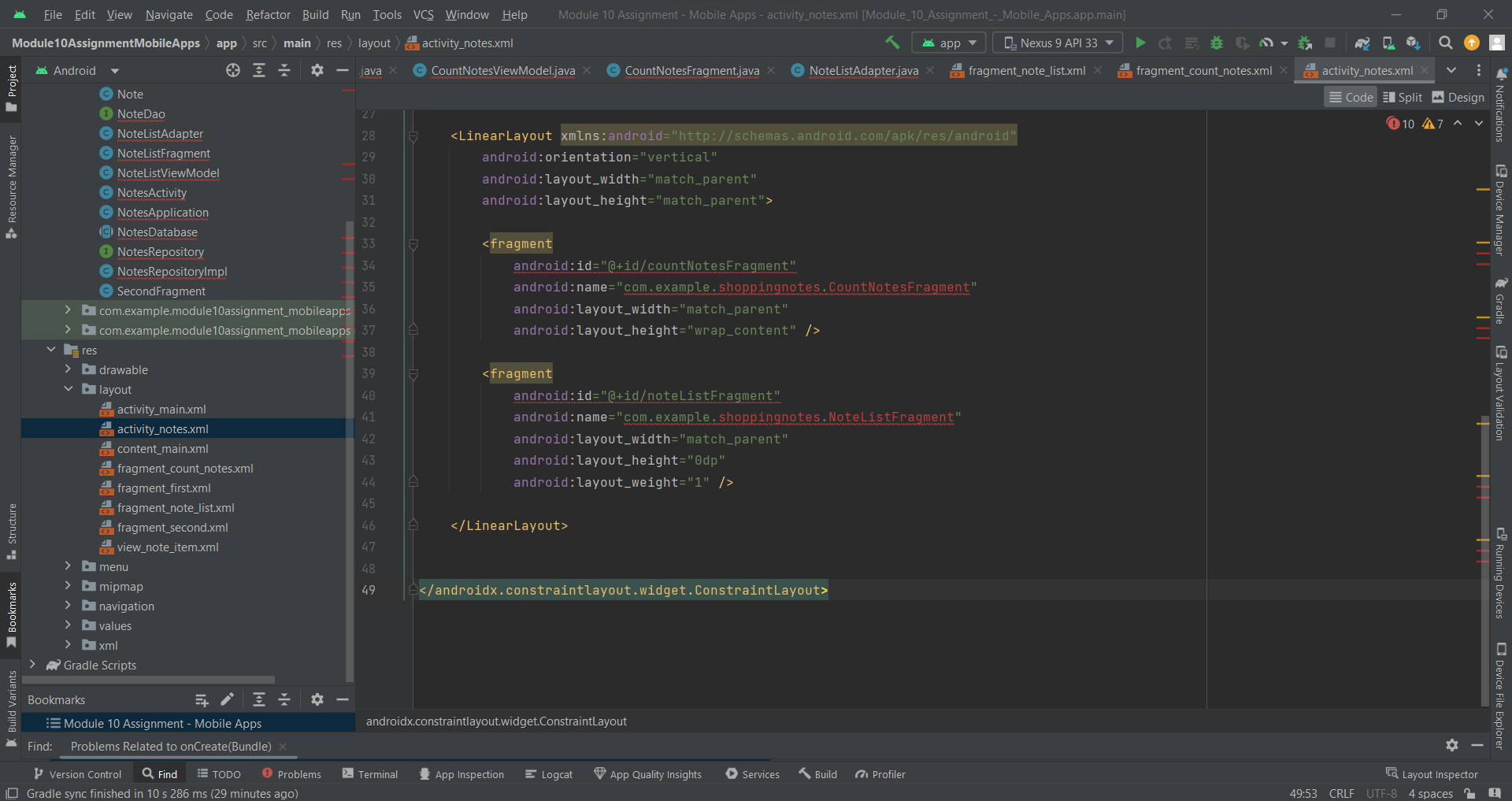
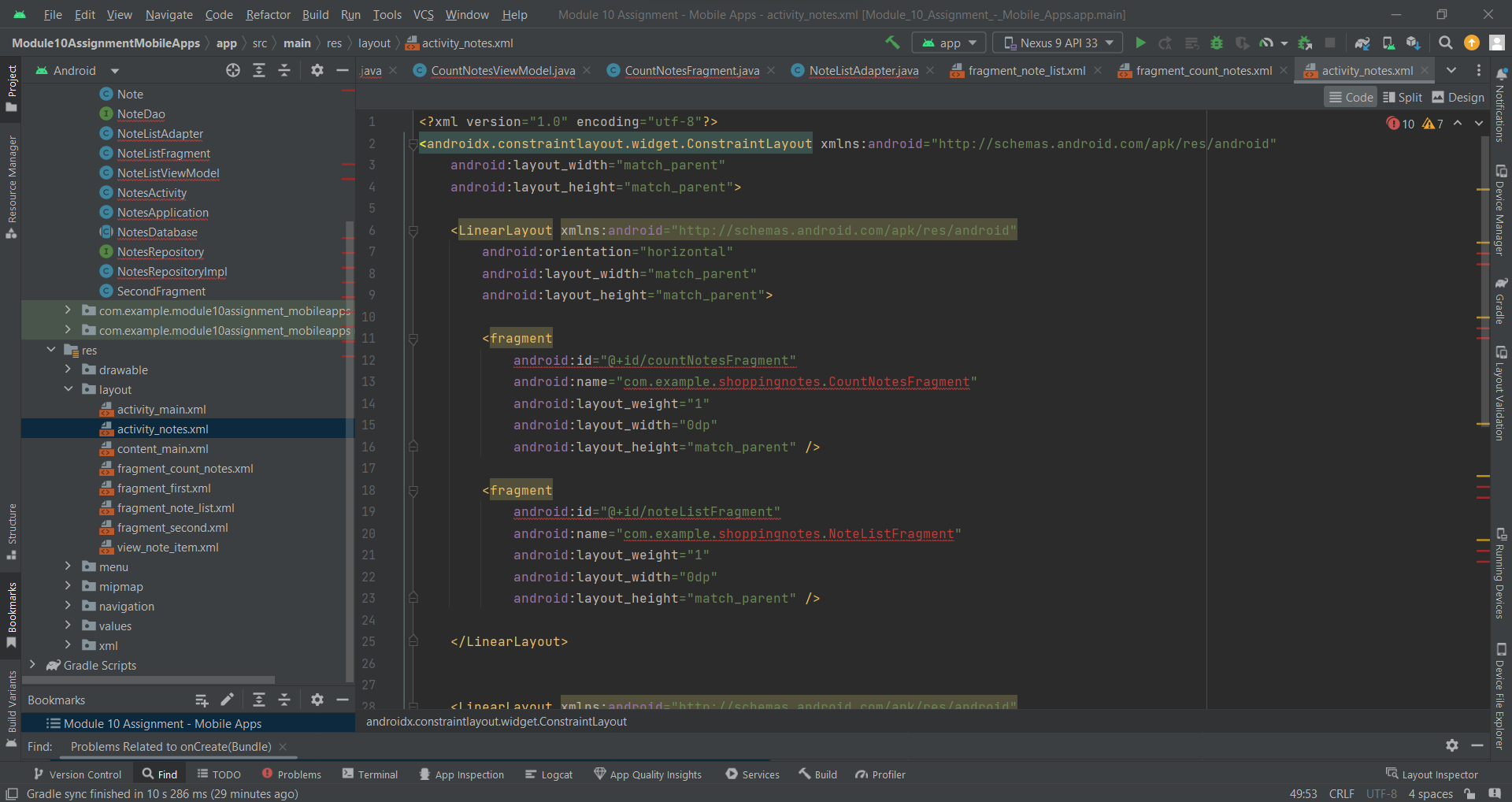
It includes a TextView to display the total number of notes, an EditText to input new note text, and a Button to add the note.



1. activity\_notes.xml (landscape and portrait mode):

These layout files define the UI for the NotesActivity in landscape and portrait modes.

They include two fragments (CountNotesFragment and NoteListFragment) arranged horizontally in landscape mode and vertically in portrait mode.



Problems during Development:

1. Room database migration issue:

One problem encountered was a database migration issue when modifying the Note entity. It required handling database schema changes and writing a migration script to preserve existing data.

1. RecyclerView performance optimization:

While implementing the NoteListFragment, a performance issue was identified with the RecyclerView when handling a large number of notes. It required implementing pagination or optimizing the RecyclerView adapter to improve the scrolling performance.

1. LiveData observation inconsistency:

There was an issue with LiveData observation inconsistency, where the UI was not always properly updated when new notes were added. It required careful management of LiveData lifecycles and ensuring proper observer registration and unregistration.

1. Fragment transaction overlapping:

During the development of the NotesActivity, there was a problem with fragment transactions overlapping when switching between landscape and portrait modes. It required reconfiguring the fragment transactions to avoid conflicts and ensure smooth transitions between orientations.

The link to Github repository is below -

<https://github.com/NRL8/Module-10-Assignment---Mobile-Apps>