Develop a notepad app with custom keypad:

* Top fragment: display the note entered.
* Bottom fragment: display several rows of letters, numbers, and symbols (each as a button) for text input; when a user pushes a button, the corresponding input is appended in the top fragment.

This assignment is about creating an app which has two fragment the top fragment and bottom fragment where the bottom fragment has the all the button with letters, numbers and symbols which when type will be displayed on the top fragment.

**MainActivity.java**: This is the app's primary class; it oversees managing the fragments, communicating between the UI and the java classes between the top and bottom fragments, and serving as the entry point for engaging with the user. When a user presses or enters a key or provides input to the bottom fragment, the MainActivity receives the input and, using the method "onkeyEntered," updates the top fragment with each new input the user inputs in the bottom fragment. I'm also utilizing several other methods, such as "onCreate," "FragmentTransaction," and "OnKeyInteraction," to help with initializing the activity, embedding the top and bottom fragments, and handling keypresses from the bottom fragment.

A screenshot of a computer

Description automatically generated

**MainActivity.xml** : In mainactivity.xml, we design the main activity's user interface. We divide the screen into two halves by using "TopFragmentContainer" for the top fragment and "BottomFragmentContainer" for the bottom fragment. We also set the size of the fragments using these methods and we use "ConstraintLayout" as the root. To achieve a responsive UI, the frame layouts are positioned and enlarged relative to one another. This configuration effectively produces a split screen within the UI in which the top and bottom fragments work independently but cooperatively within the same activity.

A screenshot of a computer

Description automatically generated

**BottomFragment.java** : The primary function of the “BottomFragment” is to create a custom keypad with in a “GridLayout” . When attached to an activity, this fragment creates an interface “OnKeyInteractionListener” for signaling key presses to the activity. The “onCreateView” method dynamically adds buttons representing letters (A-Z), numerals (0-9), and different symbols (such as semicolon, @, #, and so on) to the “GridLayout”. When a button is clicked, the linked activity is notified via "mListener.onKeyEntered" and the relevant character is sent as input. I'm generating and configuring this button with the required text and click listeners within the grid using the "addButton" method.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Bottomfragment.xml** : A 'GridLayout' is utilized within a 'ConstraintLayout' in this XML layout to create a grid-based layout . The 'GridLayout' is set to span the whole available space “match\_parent” for both width and height and has a given number of columns and rows “columnCount="4"”. The 'GridLayout' is limited to align with the boundaries of the parent 'ConstraintLayout', ensuring that it takes up the whole area of the parent layout.

A screenshot of a computer

Description automatically generated

**TopFragment.java** : this class is reponsibe for displaing text and to manage and update a “TextView” within its layout.here we ar using methods like “onCreateView”,”appendText”,”TextViewInitialization” for creaating and setting up the fragment view, to display the text in the fragment and to append “text” o the “TextView”.

A screenshot of a computer

Description automatically generated

**x** : This XML layout defines a user interface “ConstraintLayout”. It has a single “TextView” that has a fixed width 200dp and a variable height “wrap\_content”. This arrangement is commonly used to show text in a visually centered manner on a screen. The TextView is centered in the parent ConstraintLayout using constraint settings, and it includes padding around its content for improved reading and appearance.

A screenshot of a computer

Description automatically generated

Below are the screenshots of the applications

A screenshot of a phone

Description automatically generated A screenshot of a cell phone

Description automatically generated

Screenshot 1(main activity) Screenshot 2(input using the keyboard)