[2CEIT5PE5: MOBILE APPLICATION DEVELOPMENT]

Practical: 11

Aim: Create Note Android Application that can add Note, edit Note, delete Note, and set reminder date & time of note. By using Broadcast Receiver, AlarmManager set reminder of note. By using SQLite, store all notes data.

Submitted By:PATEL PURV
Enrollment number: 20012011121

Ganpat University
॥ विद्यया समाजोत्कर्षः ॥
U.V. Patel College of Engineering

Department of

AlarmBroadcastReceiver.kt

```
package com.example.madpractical11_20012021046
 import android.app.Notification import
android.app.NotificationChannel import
android.app.NotificationManager import
android.app.PendingIntent import
android.content.BroadcastReceiver import
android.content.Context import
android.content.Intent import
android.graphics.Color import
android.os.Build import android.util.Log
import androidx.core.app.NotificationCompat
import java.io.Serializable
 class AlarmBroadcastReceiver : BroadcastReceiver() { val TAG =
"AlarmBroadcastReceiver" override fun onReceive(context: Context?,
intent: Intent?) { if(intent != null && context!=null) { val note =
Note() note.id = intent.getIntExtra(Note.NOTE_ID_KEY, -1) note.title =
intent.getStringExtra(Note.NOTE_TITLE_KEY)!! note.subTitle =
intent.getStringExtra(Note.NOTE_SUBTITLE_KEY)!! note.description =
intent.getStringExtra(Note.NOTE_DESCRIPTION_KEY)!! note.modifiedTime =
intent.getStringExtra(Note.NOTE_MODIFIED_TIME_KEY)!! note.reminderTime =
intent.getLongExtra(Note.NOTE_REMINDER_TIME_KEY, 0)
note.isReminderEnable = true
```

```
Log.i(TAG, "onReceive: Note: $note")
notificationDialog( context,
NoteViewActivity::class.java,
note.title, note.description,
note )
} }
private fun notificationDialog(context: Context, cls: Class<*>, title:
String, descr: String,note: Note) {
val notificationManager =
context.getSystemService(Context.NOTIFICATION_SERVICE) as
NotificationManager val NOTIFICATION_CHANNEL_ID =
"hitensadani" if (Build.VERSION.SDK_INT >=
Build.VERSION_CODES.O) { val notificationChannel =
NotificationChannel(
NOTIFICATION_CHANNEL_ID,
"Note Application", NotificationManager.IMPORTANCE_HIGH
)
// Configure the notification channel. notificationChannel.description
= "Note Application description"
notificationChannel.enableLights(true) notificationChannel.lightColor
= Color.RED notificationChannel.vibrationPattern = longArrayOf(0,
1000, 500, 1000) notificationChannel.enableVibration(true)
notificationManager.createNotificationChannel(notificationChannel)
} val notificationIntent = Intent(context,
cls)
notificationIntent.addFlags(Intent.FLAG_ACTIV
ITY_CLEAR_TOP) //notification message will
get at NotificationView
notificationIntent.putExtra("message", "This
```

```
is a notification message")
notificationIntent.putExtra("Object", note as
Serializable) val pendingIntent =
PendingIntent.getActivity( context,
Note.REMINDER_REQUEST_CODE,
notificationIntent,
PendingIntent.FLAG_UPDATE_CURRENT or PendingIntent.FLAG_MUTABLE
)
val notificationBuilder = NotificationCompat.Builder(context,
NOTIFICATION_CHANNEL_ID) notificationBuilder.setAutoCancel(true)
.setDefaults(Notification.DEFAULT_ALL)
.setWhen(System.currentTimeMillis())
.setSmallIcon(R.mipmap.ic_launcher)
.setTicker("NoteApplication-HitenSadani")
//.setPriority(Notification.PRIORITY_MAX)
.setContentTitle(title)
.setContentText(descr)
.setContentInfo(descr + "Information")
.setAutoCancel(true)
.setContentIntent(pendingIntent) notificationManager.notify(1,
notificationBuilder.build())
}
}
```

DatabaseHelper.kt

```
package com.example.madpractical11_20012021046
 import
android.content.ContentValues import
android.content.Context import
```

```
android.database.Cursor import
android.database.sqlite.SQLiteDataba
se import
android.database.sqlite.SQLiteOpenHe
lper import java.util.ArrayList


class DatabaseHelper(context: Context?) : SQLiteOpenHelper(context,
DATABASE_NAME, null, DATABASE_VERSION) { companion object { //
Database Version private const val DATABASE_VERSION = 1
// Database Name private const val
DATABASE_NAME = "notes_db"
}
// Creating Tables override fun
onCreate(db: SQLiteDatabase) {
// create notes table db.execSQL(NotesData.CREATE_TABLE)
}
// Upgrading database
override fun onUpgrade(db: SQLiteDatabase, oldVersion: Int, newVersion:
Int) {
// Drop older table if existed db.execSQL("DROP TABLE IF
EXISTS " + NotesData.TABLE_NAME)
// Create tables again onCreate(db)
} fun insertNote(note: Note): Long
{
// get writable database as we want to write data val
db = this.writableDatabase
// insert row val id = db.insert(NotesData.TABLE_NAME, null,
getValues(note)) // close db connection db.close()
```

```
// return newly inserted row id return id }

private fun getValues(note: Note): ContentValues {

val values = ContentValues()

// `id` will be inserted automatically.

// no need to add them values.put(NotesData.COLUMN_NOTE_TITLE,

note.title) values.put(NotesData.COLUMN_NOTE_SUB_TITLE,

note.subTitle) values.put(NotesData.COLUMN_NOTE_DESCRIPTION,

note.description) values.put(NotesData.COLUMN_NOTE_REMINDER_TIME,

note.reminderTime) values.put(NotesData.COLUMN_NOTE_SET_REMINDER,

note.isReminderEnable) values.put(NotesData.COLUMN_TIMESTAMP,

note.modifiedTime) return values

} fun getNote(id: Long): Note

{

// get readable database as we are not inserting anything

val db = this.readableDatabase val cursor = db.query(

NotesData.TABLE_NAME, arrayOf(NotesData.COLUMN_ID,

NotesData.COLUMN_NOTE_TITLE,

NotesData.COLUMN_NOTE_SUB_TITLE,

NotesData.COLUMN_NOTE_DESCRIPTION,

NotesData.COLUMN_NOTE_SET_REMINDER,

NotesData.COLUMN_NOTE_REMINDER_TIME,

NotesData.COLUMN_TIMESTAMP),
NotesData.COLUMN_ID.toString() + "=?",

arrayOf(id.toString()), null, null,

null, null ) cursor?.moveToFirst() val

note = getNote(cursor)
```

```
// close the db connection

cursor!!.close() return

note

} private fun getNote(cursor: Cursor):

Note

{

// prepare note object val

note = Note(

cursor.getString(cursor.getColumnIndexOrThrow(NotesData.COLUMN_NOTE_TITLE
)),

cursor.getString(cursor.getColumnIndexOrThrow(NotesData.COLUMN_NOTE_SUB_T
ITLE)),

cursor.getString(cursor.getColumnIndexOrThrow(NotesData.COLUMN_NOTE_DESCR
IPTION)),

cursor.getString(cursor.getColumnIndexOrThrow(NotesData.COLUMN_TIMESTAMP)
) )

note.isReminderEnable =
(cursor.getInt(cursor.getColumnIndexOrThrow(NotesData.COLUMN_NOTE_SET_REM
INDER))==1)
note.reminderTime =
cursor.getLong(cursor.getColumnIndexOrThrow(NotesData.COLUMN_NOTE_REMINDE
R_TIME))

note.id =

cursor.getInt(cursor.getColumnIndexOrThrow(NotesData.COLUMN_ID)) return

note

} val allNotes:

ArrayList<Note> get() { val

notes = ArrayList<Note>()

val selectQuery = "SELECT  * FROM " + NotesData.TABLE_NAME.toString() + "
ORDER BY " +

NotesData.COLUMN_TIMESTAMP.toString() + " DESC"

val db = this.writableDatabase val cursor =
```

```
db.rawQuery(selectQuery, null) if
(cursor.moveToFirst()) { do {
notes.add(getNote(cursor))
} while (cursor.moveToNext())
}
// close db connection
db.close() return
notes
} val notesCount: Int get() { val countQuery = "SELECT  *
FROM " + NotesData.TABLE_NAME val db =
this.readableDatabase val cursor =
db.rawQuery(countQuery, null) val count = cursor.count
cursor.close()
return count } fun
updateNote(note: Note): Int { val
db = this.writableDatabase return
db.update( NotesData.TABLE_NAME,
getValues(note),
NotesData.COLUMN_ID + " = ?",
arrayOf(note.id.toString())
) } fun deleteNote(note: Note)
{ val db =
this.writableDatabase
db.delete(
NotesData.TABLE_NAME, NotesData.COLUMN_ID
+ " = ?", arrayOf(note.id.toString())
) db.close()
}}
```

MainActivity.kt:

```
package com.example.madpractical11_20012021046

 import android.os.Build import

androidx.appcompat.app.AppCompatActivity import

android.os.Bundle import android.util.Log

import android.view.LayoutInflater import

android.widget.Toast

import androidx.appcompat.app.AlertDialog import

androidx.core.view.WindowCompat import

androidx.recyclerview.widget.DefaultItemAnimator import

androidx.recyclerview.widget.LinearLayoutManager import

androidx.recyclerview.widget.RecyclerView

import
com.example.madpractical11_20012021046.databinding.ActivityMainBinding

import
com.example.madpractical11_20012021046.databinding.CustomDialogViewBindin
g

//import
com.example.madpractical11_20012021046.databinding.NoteViewDesignBinding

//import
com.example.madpractical11_20012021046.databinding.ActivityMainBinding
import java.util.ArrayList

 class MainActivity : AppCompatActivity() {

private lateinit var binding: ActivityMainBinding

private val TAG = "MainActivity"

private var listener: ((note: Note, baseListAdapter: NotesAdapter, mode:
NoteMode, position: Int)->Unit)? =

{ note: Note, _: NotesAdapter, noteMode: NoteMode, pos: Int ->

note.modifiedTime = Note.getCurrentDateTime() if (noteMode ==

NoteMode.add) { if (!createNote(note))
```

```
Toast.makeText(this, "Enter Valid Note", Toast.LENGTH_SHORT).show()
} else if (noteMode == NoteMode.edit) {
Log.i(TAG, "listener: Note:$note") if
(!updateNote(note, pos))
Toast.makeText(this, "Enter Valid Note", Toast.LENGTH_SHORT).show()
}
}
lateinit var db: DatabaseHelper private val notesList:
ArrayList<Note> = ArrayList<Note>() lateinit var
notesRecycleAdapter: NotesAdapter override fun
onCreate(savedInstanceState: Bundle?) {
WindowCompat.setDecorFitsSystemWindows(window, false)
super.onCreate(savedInstanceState)
 binding =
ActivityMainBinding.inflate(layoutInflater)
setContentView(binding.root)
 db = DatabaseHelper(this)
notesList.addAll(db.allNotes)
 notesRecycleAdapter = NotesAdapter(this,
notesList)
val mLayoutManager: RecyclerView.LayoutManager =
LinearLayoutManager(applicationContext)
binding.recyclerView.layoutManager = mLayoutManager
binding.recyclerView.itemAnimator = DefaultItemAnimator()
binding.recyclerView.setHasFixedSize(true) binding.recyclerView.adapter
= notesRecycleAdapter
 binding.addNote.setOnClickListener
{ showAlertDialog(
NoteMode.add, "Add Note",
```

```
Note("", "", "", Note.getCurrentDateTime()), -1, notesRecycleAdapter
)}} private fun createNote(note:
Note):Boolean { if(!note.isValid()) return
false
val id = db.insertNote(note) val n: Note = db.getNote(id)
notesList.add(0, n)
notesRecycleAdapter.notifyItemInserted(0)
note.saveNote(this) return true } private fun
updateNote(note: Note, position: Int):Boolean {
if(!note.isValid()) return false val n: Note =
notesList[position]
n.changeValue(note)
Log.i(TAG, "updateNote: note:: $n")
db.updateNote(n) notesList[position] = n
notesRecycleAdapter.notifyItemChanged(position)
note.saveNote(this) return true } fun
deleteNote(position: Int) {
db.deleteNote(notesList[position])
notesList.removeAt(position)
notesRecycleAdapter.notifyItemRemoved(position)
} fun
showAlertDialog(
mode:NoteMode,
dialogTitle: String,
note: Note,
position: Int,
baseListAdapter:
NotesAdapter
```

```
) { val builder: AlertDialog.Builder = AlertDialog.Builder(this)
builder.setTitle(dialogTitle) val binding =
CustomDialogViewBinding.inflate(LayoutInflater.from(this))
binding.ntTitle.setText(note.title)
binding.ntSubTitle.setText(note.subTitle)
binding.ntDescription.setText(note.description)
binding.reminderSwitch.isChecked = note.isReminderEnable if
(Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
binding.reminderTime.hour = note.getHour() binding.reminderTime.minute =
note.getMinute()
} else{ binding.reminderTime.currentHour =
note.getHour() binding.reminderTime.currentMinute =
note.getMinute()
} builder.setView(binding.root)
builder.setPositiveButton(
"OK" ) { _, _ -> val newNote = Note(note) newNote.title =
binding.ntTitle.text.toString() newNote.subTitle =
binding.ntSubTitle.text.toString() newNote.description =
binding.ntDescription.text.toString()
newNote.isReminderEnable = binding.reminderSwitch.isChecked
 if (Build.VERSION.SDK_INT >=
Build.VERSION_CODES.M) newNote.reminderTime =
Note.getMillis(binding.reminderTime.hour,
binding.reminderTime.minute) else
newNote.reminderTime = Note.getMillis(binding.reminderTime.currentHour,
binding.reminderTime.currentMinute)
Log.i(TAG, "showAlertDialog: OK Button:: Note:$newNote")
listener?.invoke(newNote, baseListAdapter, mode, position)
```

```
} val dialog: AlertDialog =
builder.create() dialog.show()
}}
```

Note.kt

```
package com.example.madpractical11_20012021046
 import android.app.AlarmManager import
android.app.PendingIntent import
android.content.Context import
android.content.Context.ALARM_SERVICE import
android.content.Intent import
android.os.Parcel import
android.os.Parcelable import
java.io.Serializable import
java.text.DateFormat import
java.text.SimpleDateFormat import
java.util.* enum class NoteMode{ edit, add }
class Note(var title:String, var subTitle:String, var description:String,
var modifiedTime:String, var isReminderEnable: Boolean =
false):Serializable
{ var reminderTime:Long =
System.currentTimeMillis() var id =
noteIdGeneration() set(value) { field = value
if(idNote < value) idNote = value
}
constructor():this("","","","")
{    }
constructor(note: Note) :
this(note.title,note.subTitle,note.description,note.modifiedTime,note.isR
eminderEnable) { reminderTime = note.reminderTime
```

```
} fun isValid() :Boolean{ if(title.isEmpty()
|| description.isEmpty()) return false
return true
} fun changeValue(newValue:
Note)
{ title = newValue.title subTitle =
newValue.subTitle description =
newValue.description modifiedTime =
newValue.modifiedTime isReminderEnable =
newValue.isReminderEnable reminderTime =
newValue.reminderTime

} fun getReminderText()
:String
{
return "Reminder: "+(SimpleDateFormat("MMM, dd yyyy hh:mm a") as
DateFormat).format(
Date(reminderTime)
) } fun saveNote(context:
Context)
{ if(isReminderEnable)
{ setReminder(context,this)
}} fun getHour():Int{ val cal =
Calendar.getInstance() cal.time
= Date(reminderTime) return
cal[Calendar.HOUR_OF_DAY]
} fun getMinute():Int{ val cal =
Calendar.getInstance() cal.time
```

```
= Date(reminderTime) return

cal[Calendar.MINUTE]

} fun

calcReminder()

{ if(reminderTime <

System.currentTimeMillis()) isReminderEnable

= false

}
override fun toString(): String {

return "$id\n"+title +"\n"+subTitle
+"\n"+description+"\nReminder:$isReminderEnable" +"\n"+getReminderText()

} companion object { var

idNote = 0 fun

noteIdGeneration():Int

{ idNote++

return idNote

} val REMINDER_REQUEST_CODE = 1000 val

NOTE_ID_KEY = "Id" val NOTE_TITLE_KEY =

"Title" val NOTE_SUBTITLE_KEY = "SubTitle"

val NOTE_DESCRIPTION_KEY = "Description"

val NOTE_MODIFIED_TIME_KEY = "ModifiedTime"

val NOTE_REMINDER_TIME_KEY = "ReminderTime"

 fun getCurrentDateTime(): String { val cal =

Calendar.getInstance() val df: DateFormat =

SimpleDateFormat("MMM, dd yyyy hh:mm:ss a") return

df.format(cal.time)

} fun

getMillis(hour:Int,min:Int):Long
```

```
{ val setcalendar =
Calendar.getInstance()
setcalendar[Calendar.HOUR_OF_DAY] = hour
setcalendar[Calendar.MINUTE] = min
setcalendar[Calendar.SECOND] = 0 return
setcalendar.timeInMillis
}  fun setReminder(context: Context, note: Note) { val intent =
Intent(context, AlarmBroadcastReceiver::class.java)
intent.putExtra(NOTE_ID_KEY, note.id)
intent.putExtra(NOTE_TITLE_KEY, note.title)
intent.putExtra(NOTE_SUBTITLE_KEY, note.subTitle)
intent.putExtra(NOTE_DESCRIPTION_KEY, note.description)
intent.putExtra(NOTE_MODIFIED_TIME_KEY, note.modifiedTime)
intent.putExtra(NOTE_REMINDER_TIME_KEY, note.reminderTime)
 val pendingIntent
=
PendingIntent.getBroadcast(context, note.id, intent,
PendingIntent.FLAG_MUTABLE or PendingIntent.FLAG_CANCEL_CURRENT)
val alarmManager = context.getSystemService(ALARM_SERVICE) as
AlarmManager
 if(note.isReminderEnable)
{ alarmManager.setExact(
AlarmManager.RTC_WAKEUP,
note.reminderTime,
pendingIntent
) } else
alarmManager.cancel(pendingIntent)
}
}
```

```
}
```

NoteViewActivity.kt

```
package com.example.madpractical11_20012021046
 import
androidx.appcompat.app.AppCompatActivity import
android.os.Bundle import android.util.Log
import android.view.View import
androidx.core.view.WindowCompat
import
com.example.madpractical11_20012021046.databinding.ActivityNoteViewBindin
g  class NoteViewActivity : AppCompatActivity() { private lateinit var
binding: ActivityNoteViewBinding private lateinit var note:Note override
fun onCreate(savedInstanceState: Bundle?) {
WindowCompat.setDecorFitsSystemWindows(window, false)
super.onCreate(savedInstanceState) note =
intent.getSerializableExtra("Object") as Note binding =
ActivityNoteViewBinding.inflate(layoutInflater)
setContentView(binding.root)
 with(note){ binding.noteTitle.text =
this.title binding.noteSubtitle.text =
this.subTitle binding.noteDescription.text =
this.description binding.noteDate.text =
this.modifiedTime this.calcReminder()
if(this.isReminderEnable)
{ binding.noteReminder.visibility = View.VISIBLE
binding.noteReminder.text = this.getReminderText()
} else binding.noteReminder.visibility =
View.GONE
```

```
}}}
```

NotesAdapter.kt

```
package com.example.madpractical11_20012021046

 import android.content.Context import

android.content.Intent import

android.os.Parcelable import

android.view.LayoutInflater import

android.view.View import android.view.ViewGroup

import androidx.recyclerview.widget.RecyclerView

import
com.example.madpractical11_20012021046.databinding.NoteViewDesignBinding
import java.io.Serializable


class NotesAdapter (private val context: Context, private val
array:ArrayList<Note>):

RecyclerView.Adapter<NotesAdapter.NotesViewHolder>() {

inner class NotesViewHolder(val binding: NoteViewDesignBinding):
RecyclerView.ViewHolder(binding.root)


override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
NotesViewHolder {
val binding =
NoteViewDesignBinding.inflate(LayoutInflater.from(parent.context),

parent, false) return NotesViewHolder(binding)

}  override fun onBindViewHolder(holder: NotesViewHolder, position:

Int) { with(holder){ with(array[position]){ binding.noteTitle.text =

this.title binding.noteSubTitle.text = this.subTitle

binding.noteContent.text = this.description binding.noteReminder.text =

this.modifiedTime val obj = this as Serializable this.calcReminder()

if(this.isReminderEnable)
```

```
{ binding.noteReminder.visibility = View.VISIBLE
binding.noteReminder.text = this.getReminderText()
} else binding.noteReminder.visibility =
View.GONE
binding.deleteNote.setOnClickListener{
(context as MainActivity).deleteNote(position)
} binding.cardView.setOnClickListener
{
Intent(this@NotesAdapter.context, NoteViewActivity::class.java).apply {
putExtra("Object",obj) this@NotesAdapter.context.startActivity(this)
}
} binding.editNote.setOnClickListener
{
(context as MainActivity).showAlertDialog(
NoteMode.edit,
"Edit Note",
this, position,
this@NotesAdapter
)
}
}
}  override fun getItemCount():
Int { return array.size
}
```

} **NotesData.kt**

```
package com.example.madpractical11_20012021046
```

```
 class NotesData { companion
object{ const val TABLE_NAME =
"notes"
 const val COLUMN_ID = "id" const val
COLUMN_NOTE_TITLE = "note_title" const val
COLUMN_NOTE_SUB_TITLE = "note_sub_title" const val
COLUMN_NOTE_DESCRIPTION = "note_description" const val
COLUMN_NOTE_SET_REMINDER = "note_set_reminder" const
val COLUMN_NOTE_REMINDER_TIME =
"note_set_reminder_time" const val COLUMN_TIMESTAMP =
"note_modified_timestamp" val CREATE_TABLE = ("CREATE
TABLE " + TABLE_NAME + "("
+ COLUMN_ID + " INTEGER PRIMARY KEY AUTOINCREMENT,"
+ COLUMN_NOTE_TITLE + " TEXT,"
+ COLUMN_NOTE_SUB_TITLE + " TEXT,"
+ COLUMN_NOTE_DESCRIPTION + " TEXT,"
+ COLUMN_NOTE_SET_REMINDER + " INTEGER,"
+ COLUMN_NOTE_REMINDER_TIME + " INTEGER,"
+ COLUMN_TIMESTAMP + " TEXT"
+ ")")
}
}
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
package="com.example.madpractical11_20012021046">
```

```
<uses-permission android:name="android.permission.SCHEDULE_EXACT_ALARM"/>
<application android:allowBackup="true"
android:dataExtractionRules="@xml/data_extraction_rules"
android:fullBackupContent="@xml/backup_rules"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:roundIcon="@mipmap/ic_launcher_round"
android:supportsRtl="true"
android:theme="@style/Theme.MADPractical11_20012021046"
tools:targetApi="31">
<receiver
android:name="com.example.madpractical11_20012021046.AlarmBroadcastReceiv
er" android:directBootAware="true" android:enabled="true"
android:exported="true">
<intent-filter>
<action android:name="android.intent.action.BOOT_COMPLETED"/>
<action android:name="android.intent.action.LOCKED_BOOT_COMPLETED"/>
</intent-filter>
</receiver> <activity
android:name="com.example.madpractical11_20012021046.NoteViewActivity"
android:exported="false" />
<activity
android:name="com.example.madpractical11_20012021046.MainActivity"
android:exported="true">
<intent-filter>
<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />
```

```
</intent-filter>
</activity>
</application>

</manifest>
```

activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical" tools:context=".MainActivity">

<androidx.recyclerview.widget.RecyclerView
android:id="@+id/recyclerView"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_marginTop="50dp"
tools:listitem="@layout/note_view_design" tools:itemCount="5"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"/>

<com.google.android.material.floatingactionbutton.FloatingActionButton
android:id="@+id/addNote" android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
android:layout_gravity="bottom|right" android:layout_marginEnd="25dp"
android:layout_marginBottom="70dp"
android:src="@drawable/ic_baseline_add_24"
app:elevation="7dp" app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintBottom_toBottomOf="parent"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

activity_note_view.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent" android:layout_height="match_parent"
android:orientation="vertical" tools:context=".NoteViewActivity">
<TextView android:id="@+id/note_title"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="TextView"
android:textStyle="bold"
android:textSize="18dp"
android:layout_marginStart="15dp"
android:layout_marginTop="60dp"/>
<TextView android:id="@+id/note_subtitle"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="TextView"
android:layout_marginStart="15dp" />
```

```
<TextView android:id="@+id/note_description"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="TextView"
android:textSize="15dp"
android:layout_marginStart="15dp"
android:layout_marginTop="5dp"/>
<TextView android:id="@+id/note_date"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="TextView"
android:textSize="12dp"
android:layout_marginStart="15dp"
android:layout_marginTop="7dp"/>
<TextClock android:id="@+id/note_reminder"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:textSize="12dp"
android:layout_marginStart="15dp"
android:visibility="gone" />
</LinearLayout>
```

custom_dialog_view.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent" android:layout_height="match_parent"
xmlns:app="http://schemas.android.com/apk/res-auto">

<androidx.core.widget.NestedScrollView
android:layout_width="match_parent"
```

```
android:layout_height="match_parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintBottom_toBottomOf="parent">
<LinearLayout android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="vertical"
app:layout_constraintTop_toTopOf="parent">
<com.google.android.material.textfield.TextInputLayout
android:layout_width="match_parent" android:layout_height="wrap_content"
style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"
app:boxStrokeColor="@color/black">
<com.google.android.material.textfield.TextInputEditText
android:id="@+id/ntTitle"
android:layout_width="match_parent"
android:layout_height="match_parent" android:hint="Note
Title"
android:drawableEnd="@drawable/ic_baseline_event_note_24"
android:drawableTint="@color/purple_500"/>
</com.google.android.material.textfield.TextInputLayout>
<com.google.android.material.textfield.TextInputLayout
android:layout_width="match_parent" android:layout_height="wrap_content"
style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"
app:boxStrokeColor="@color/black">
<com.google.android.material.textfield.TextInputEditText
android:id="@+id/ntSubTitle"
android:layout_width="match_parent"
```

```
android:layout_height="match_parent" android:hint="Note
Sub Title"
android:drawableEnd="@drawable/ic_baseline_event_note_24"
android:drawableTint="@color/purple_500"/>
</com.google.android.material.textfield.TextInputLayout>
<com.google.android.material.textfield.TextInputLayout
android:layout_width="match_parent" android:layout_height="wrap_content"
style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"
app:boxStrokeColor="@color/black">
<com.google.android.material.textfield.TextInputEditText
android:id="@+id/ntDescription"
android:layout_width="match_parent"
android:layout_height="125dp" android:hint="Note
Description" android:inputType="textMultiLine"
android:drawableEnd="@drawable/ic_baseline_event_note_24"
android:drawableTint="@color/purple_500"/>
</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.switchmaterial.SwitchMaterial
android:id="@+id/reminderSwitch" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:layout_gravity="right"
android:layout_marginEnd="10dp" android:gravity="end|center_vertical"
android:textSize="30sp" android:text="Set Reminder"/> <TimePicker
android:id="@+id/reminderTime" android:layout_width="match_parent"
android:layout_height="wrap_content"/>
</LinearLayout>
</androidx.core.widget.NestedScrollView>
```

</androidx.constraintlayout.widget.ConstraintLayout> **note_view_design.xml:**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent" android:layout_height="wrap_content"
xmlns:app="http://schemas.android.com/apk/res-auto"
android:orientation="vertical" android:layout_marginHorizontal="10dp">

<com.google.android.material.card.MaterialCardView
android:id="@+id/cardView" android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_margin="10dp" app:cardElevation="7dp"
app:cardCornerRadius="10dp"> <LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="horizontal">
<LinearLayout android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:orientation="vertical"
android:layout_marginTop="15dp"
android:layout_marginBottom="15dp"
android:layout_weight="2"
android:layout_gravity="center_vertical">
<TextView
android:id="@+id/noteTitle"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Medium Text"
android:textStyle="bold"
```

```
android:textSize="18sp"
android:layout_marginLeft="10dp"
android:padding="2dp" /> <TextView
android:id="@+id/noteSubTitle"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="TextView"
android:layout_marginLeft="10dp"
android:padding="2dp" /> <TextView
android:id="@+id/noteContent"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="TextView"
android:visibility="gone"
android:layout_marginLeft="10dp"
android:padding="2dp" /> <TextView
android:id="@+id/noteDate"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="TextView"
android:visibility="gone"
android:layout_marginLeft="10dp"
android:padding="2dp" /> <TextView
android:id="@+id/noteReminder"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="TextView"
android:visibility="gone"
```

```
android:layout_marginLeft="10dp"
android:padding="2dp" />
</LinearLayout> <ImageView
android:id="@+id/editNote"
android:layout_width="wrap
_content"
android:layout_height="wra
p_content"
android:src="@drawable/ic_
baseline_edit_24"
android:layout_weight="1"
app:tint="@color/green"
android:layout_gravity="ce
nter_vertical"
android:elevation="7dp"/>
<ImageView android:id="@+id/deleteNote"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:src="@drawable/ic_baseline_delete_24"
android:layout_weight="1"
android:layout_gravity="center_vertical"
app:tint="@color/red"
android:elevation="7dp"/>
</LinearLayout>
</com.google.android.material.card.MaterialCardView>

</LinearLayout>
```

Practical: 11

Output: