

Android----记事本

软件名称：

记事本

完成人：

谭增明

学号：

20192131002

完成时间：

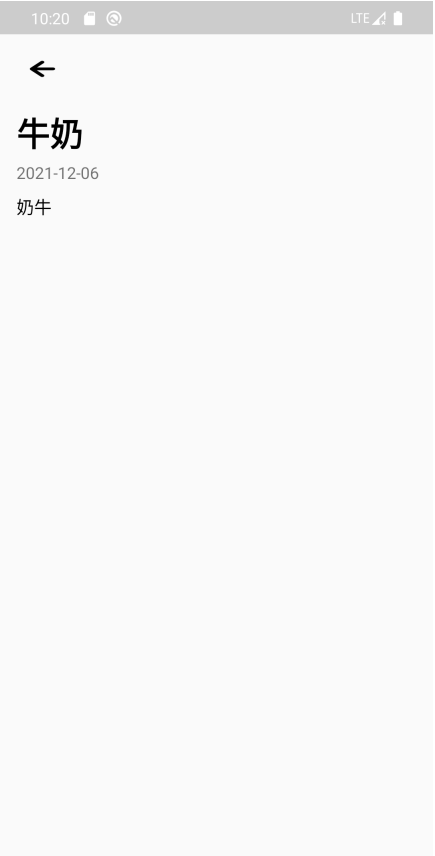
2021-12-06

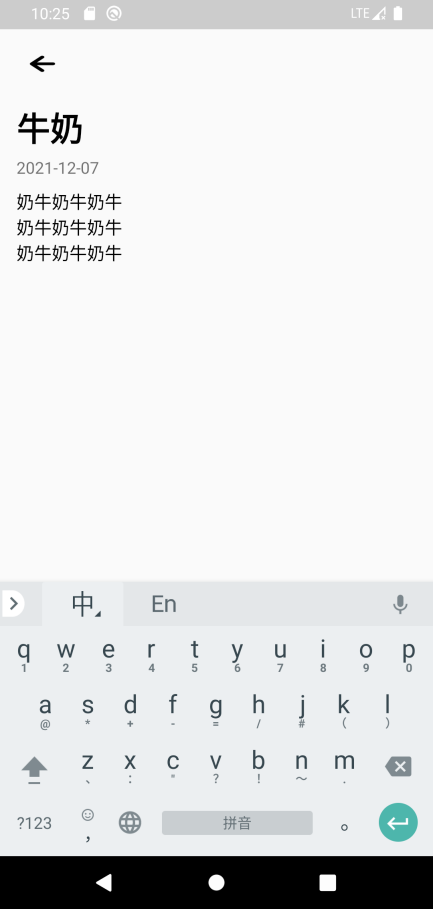
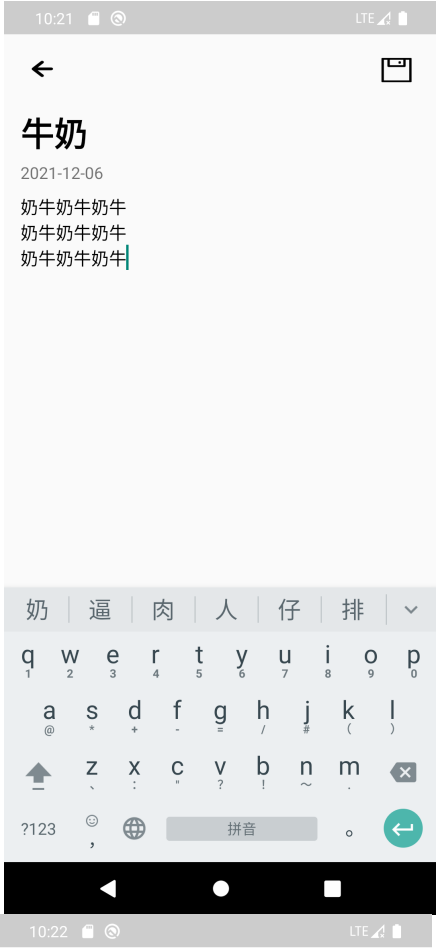
界面设计：

1.切换视图



2.修改内容





3.搜索阴影



4.添加内容



5.长按删除

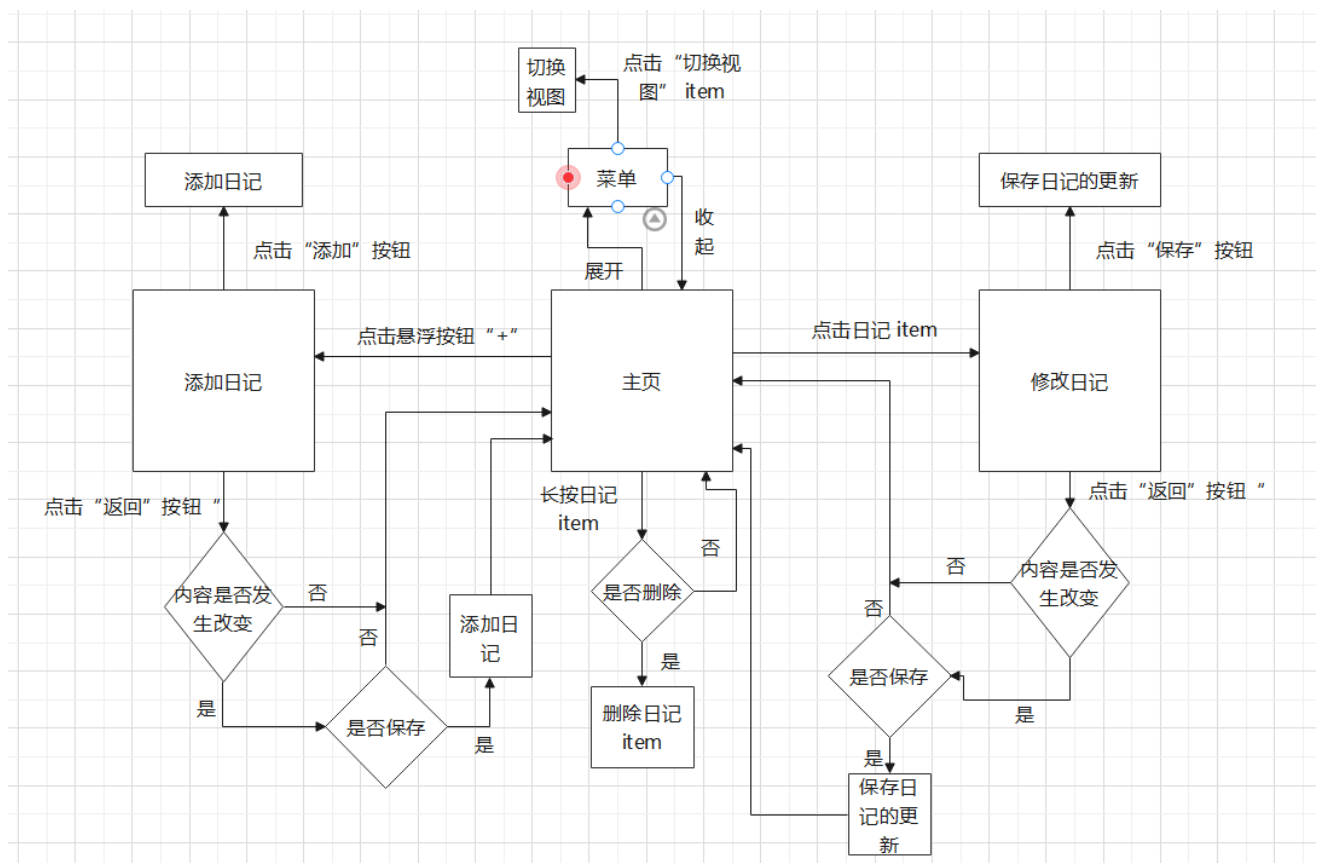


软件操作流程：

文字流程：

1. 点击悬浮按钮，可输入内容进行添加日记
2. 展开popupmenu，可选择在列表视图、宫格视图间切换
3. 长按某条日记，会弹出对话框提示是否选择删除
4. 搜索框内输入，会根据输入内容在所有日记中的title和content里进行过滤
5. 当焦点在搜索框时，UI背景会更改
6. 点击某条日记后，可对日记标题和内容进行修改，如果内容发生了变化且还未保存就要退出系统会提示是否需要保存

流程图流程：



难点：

1. 去掉标题栏的同时在右上角设置一个menu展开按钮
2. 搜索内容的过滤
3. 圆角的LinearLayout设置
4. 搜索框的焦点问题

解决方案：

1. 通过popupmenu
2. 在NoteAdapter里设置一个过滤器
3. 开始时是想通过先构造一个圆角的rectangle的shape，然后背景设置为它就好了，但是发现这样子那个圆角的矩形的边太粗台粗了，就算修改rectangle.xml来调整边框的边的粗细也无法生效，最后是通过在item的外层LinearLayout的外面再套一层


```
1 | <androidx.cardview.widget.CardView >
2 | </androidx.cardview.widget.CardView>
```

设置CardView的 app:cardCornerRadius="15dp" 来设置圆角

4. 通过键盘的收放来判断焦点是否在搜索框里

不足之处:

1. 菜单里没有批量删除功能
2. 菜单里没有选择排序方式功能

今后设想:

1. 增加批量删除功能
2. 增加排序方式功能

代码设计:

MainActivity.kt

```
1 | package com.example.note_two
2 |
3 | import android.annotation.SuppressLint
4 | import android.content.Intent
5 | import android.graphics.Color
6 | import android.media.AudioManager
7 | import androidx.appcompat.app.AppCompatActivity
8 | import android.os.Bundle
9 | import android.util.Log
10 | import android.view.inputmethod.EditorInfo
11 | import androidx.recyclerview.widget.DividerItemDecoration
12 | import androidx.recyclerview.widget.LinearLayoutManager
13 | import kotlinx.android.synthetic.main.activity_main.*
14 | import android.text.TextUtils
15 | import android.view.View
16 | import android.view.Window
17 | import android.view.WindowManager
18 | import android.widget.*
19 | import androidx.recyclerview.widget.StaggeredGridLayoutManager
20 |
21 | import com.google.android.material.snackbar.Snackbar
22 | import kotlinx.android.synthetic.main.add_note.*
23 | import java.text.SimpleDateFormat
24 | import java.util.*
25 | import kotlin.collections.ArrayList
26 | import java.util.*
27 |
28 | public var search_text:String = ""
29 | public var IS_ChANGED:Int=1
30 | public var title_ = ""
31 | public var content_ = ""
32 | public var time_ = ""
33 | public var id_=-1
34 | public var IS_INSERT=0
35 | class MainActivity : AppCompatActivity() {
36 |     private var g_l_choice=0
37 |
38 |     private val noteList = ArrayList<Note>()
39 |     var adapter = NoteAdapter(noteList)
```

```

40     @SuppressWarnings("ResourceType")
41     override fun onCreate(savedInstanceState: Bundle?) {
42         super.onCreate(savedInstanceState)
43         setContentView(R.layout.activity_main)
44
45
46         window.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SYSTEM_BAR_BACKGROUNDS);
47         window.setStatusBarColor(this.resources.getColor(R.color.back))
48         supportActionBar?.hide()
49         searchView.setImeOptions(EditorInfo.IME_ACTION_SEARCH)
50         searchView.isSubmitButtonEnabled = true
51         searchView.isFocusable = true
52         recyclerView.addItemDecoration(SpacesItemDecoration(40))
53         menu_activity.setOnClickListener {
54             val popupMenu: PopupMenu = PopupMenu(this, menu_activity)
55             popupMenu.menuInflater.inflate(R.menu.menu_title, popupMenu.menu)
56
57             popupMenu.setOnMenuItemClickListener(PopupMenu.OnMenuItemClickListener { item
58                 ->
59                     when(item.itemId) {
60                         R.id.action_crack -> {
61                             g_l_choice += 1
62                             g_l_choice %= 2
63                             if(g_l_choice==1){
64                                 val layoutManager_ =
65                                     StaggeredGridLayoutManager(2, StaggeredGridLayoutManager.VERTICAL)
66                                 recyclerView.layoutManager = layoutManager_
67                                 adapter = NoteAdapter(noteList)
68                                 recyclerView.adapter = adapter
69                                 adapter.filter.filter("")
70                             } else{
71                                 val layoutManager_ = LinearLayoutManager(this)
72                                 recyclerView.layoutManager = layoutManager_
73                                 adapter = NoteAdapter(noteList)
74                                 recyclerView.adapter = adapter
75                                 adapter.filter.filter("")
76                             }
77                         }
78                     }
79                 })
80             popupMenu.show()
81         }
82
83         fab_plus.setOnClickListener {
84             val intent=Intent(this,add_note::class.java)
85             startActivity(intent)
86         }
87         title_activity.onFocusChangeListener = View.OnFocusChangeListener { v,
88             hasFocus ->
89                 if (hasFocus) {
90
91                     main_xml.setBackgroundColor(this.resources.getColor(R.color.white))
92                 } else {
93
94                     main_xml.setBackgroundColor(this.resources.getColor(R.color.back))
95                 }
96             }
97     }
98 }

```

```

94
95     private fun init(){
96         noteList.clear()
97         val dbHelper = MyDatabaseHelper(this, "Note.db", 1)
98         val db = dbHelper.writableDatabase
99         var cursor = db.query("Note",null,null,null,null,null, "time")
100         while(cursor.moveToNext()){
101             val id = cursor.getInt(cursor.getColumnIndex("id"))
102             val title = cursor.getString(cursor.getColumnIndex("title"))
103             val content = cursor.getString(cursor.getColumnIndex("content"))
104             val time = cursor.getString(cursor.getColumnIndex("time"))
105             if(time.length<10){
106                 time="2021-12-06"
107             }
108             var con:String = time.subSequence(0,10).toString()
109             con += " | "
110             con += content
111             noteList.add(Note(title,con,id))
112         }
113         db.close()
114     }
115
116
117     override fun onResume() {
118         super.onResume()
119
120
121
122
123         if(g_l_choice==1){
124             val layoutManager =
StaggeredGridLayoutManager(2,StaggeredGridLayoutManager.VERTICAL)
125             recyclerView.layoutManager = layoutManager
126         } else{
127             val layoutManager = LinearLayoutManager(this)
128             recyclerView.layoutManager = layoutManager
129         }
130
131         if(IS_ChANGED!=0){
132             IS_ChANGED=0
133
134             val dbHelper = MyDatabaseHelper(this, "Note.db", 1)
135             val db = dbHelper.writableDatabase
136             db.execSQL(
137                 "update Note set title=?,content=?,time=? where id=?",
138                 arrayOf(title_, content_, time_, id_+1)
139             )
140             db.close()
141
142
143             init()
144             adapter = NoteAdapter(noteList)
145             recyclerView.adapter = adapter
146         }
147         if(IS_INSERT==1){
148             IS_INSERT=0
149             init()
150             adapter = NoteAdapter(noteList)
151             recyclerView.adapter = adapter
152         }
153         else if(IS_INSERT==2){

```

```

154         IS_INSERT=0
155         val dbHelper = MyDatabaseHelper(this, "Note.db", 1)
156         val db = dbHelper.writableDatabase
157         val cursor = db.rawQuery("select count(id) from Note", null)
158         cursor.moveToFirst()
159         val number=cursor.getInt(0)
160         db.execSQL("insert into Note (id,title,content,time)
values(?,?,?,?)", arrayOf(number+1,title_,
161             content_, time_))
162         db.close()
163         dbHelper.close()
164         init()
165         adapter = NoteAdapter(noteList)
166         recyclerView.adapter = adapter
167     }
168     adapter.filter.filter(search_text)
169
170     // 设置搜索文本监听
171     searchView.setOnQueryTextListener(object :
SearchView.OnQueryTextListener {
172         // 当点击搜索按钮时触发该方法
173         override fun onQueryTextSubmit(query: String): Boolean {
174             adapter.filter.filter(query)
175             search_text = query
176             Log.d("app1123", search_text)
177             searchView.clearFocus()
178             recyclerView.requestFocus()
179             return false
180         }
181
182         // 当搜索内容改变时触发该方法
183         override fun onQueryTextChange(newText: String): Boolean {
184             //do something
185             //当没有输入任何内容的时候清除结果，看实际需求
186             search_text = newText
187             if (TextUtils.isEmpty(newText)){
188                 adapter.filter.filter("")
189             } else{
190                 adapter.filter.filter(newText)
191             }
192             return false
193         }
194     })
195
196     clearFource()
197     SoftKeyBoardListener.setListener(this, object :
SoftKeyBoardListener.OnSoftKeyBoardChangeListener {
198         override fun keyBoardShow(height: Int) {}
199         override fun keyBoardHide(height: Int) {
200             clearFource()
201         }
202     })
203
204     private fun clearFource() {
205         if (searchView != null) {
206             searchView.clearFocus()
207         }
208     }
209
210     private fun clearFource() {
211         if (searchView != null) {
212             searchView.clearFocus()
213         }
214     }

```

```

213         title_activity.setFocusable(true)
214         title_activity.setFocusableInTouchMode(true)
215         title_activity.requestFocus()
216     }
217
218 }

```

add_change.kt

```

1  package com.example.note_two
2
3  import android.content.DialogInterface
4  import androidx.appcompat.app.AppCompatActivity
5  import android.os.Bundle
6  import android.view.View
7  import android.view.WindowManager
8  import androidx.appcompat.app.AlertDialog
9  import kotlinx.android.synthetic.main.add_change.*
10 import java.util.*
11
12 class add_change : AppCompatActivity() {
13     public var IS_changeded: Int = 0
14     override fun onCreate(savedInstanceState: Bundle?) {
15         super.onCreate(savedInstanceState)
16         setContentView(R.layout.add_change)
17         supportActionBar?.hide()
18
19         window.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SYSTEM_BAR_BACKGROUNDS);
20         window.setStatusBarColor(this.resources.getColor(R.color.back))
21
22         val id_num: Int = intent.getIntExtra("id_num", 0)
23         val title_ch = intent.getStringExtra("title")
24         val content_ch = intent.getStringExtra("content")
25         val time_ch = intent.getStringExtra("time")
26         edit_title.setText(title_ch)
27         content_edit.setText(content_ch)
28         dateText.setText(time_ch)
29         dateText.clearFocus()
30         content_edit.clearFocus()
31         edit_title.clearFocus()
32         edit_title.onFocusChangeListener = View.OnFocusChangeListener { v,
hasFocus ->
33             if (hasFocus) {
34                 // 此处为得到焦点时的处理内容
35                 save.setVisibility(View.VISIBLE);
36                 IS_changeded = 1
37             } else {
38                 IS_changeded -= 1
39                 save.setVisibility(View.GONE);
40                 // 此处为失去焦点时的处理内容
41             }
42         }
43
44         content_edit.onFocusChangeListener = View.OnFocusChangeListener { v,
hasFocus ->
45             if (hasFocus) {
46                 save.setVisibility(View.VISIBLE);
47                 IS_changeded = 1
48                 // 此处为得到焦点时的处理内容

```

```

48         } else {
49             save.setVisibility(View.GONE);
50             // 此处为失去焦点时的处理内容
51         }
52     }
53
54     save.setOnClickListener {
55         IS_ChANGED=1
56         IS_changed=0
57         title_ = edit_title.text.toString()
58         time_ = DateToString.DateToString_(Date()).subSequence(0,
10).toString()
59         content_ = content_edit.text.toString()
60         edit_title.clearFocus()
61         content_edit.clearFocus()
62
63         dateText.setText(DateToString.DateToString_(Date()).subSequence(0,10).toString(
64         ))
65         dateText.clearFocus()
66     }
67
68     return_change.setOnClickListener {
69         if(IS_changed==0){
70             finish()
71         } else{
72             AlertDialog.Builder(this).apply {
73                 setMessage("是否保存更改")
74                 setPositiveButton("保存") {
75                     dialog, which->
76
77                         title_ = edit_title.text.toString()
78                         time_ =
79                         DateToString.DateToString_(Date()).subSequence(0, 10).toString()
80                         content_ = content_edit.text.toString()
81                         IS_ChANGED=1
82                         finish()
83                     }
84                 setNegativeButton("不保存") {
85                     dialog, which->
86                         finish()
87                     }
88                 show()
89             }
90         }
91     }
92 }

```

add_note.kt

```

1 package com.example.note_two
2
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5 import android.view.View
6 import android.view.WindowManager
7 import androidx.appcompat.app.AlertDialog
8 import com.example.note_two.SoftKeyBoardListener.OnSoftKeyBoardChangeListener
9 import kotlinx.android.synthetic.main.add_change.*

```

```

10 import kotlinx.android.synthetic.main.add_note.*
11 import kotlinx.android.synthetic.main.add_note.content_edit
12 import kotlinx.android.synthetic.main.add_note.dateText
13 import kotlinx.android.synthetic.main.add_note.edit_title
14 import kotlinx.android.synthetic.main.add_note.return_change
15 import kotlinx.android.synthetic.main.add_note.save
16 import java.util.*
17
18 class add_note : AppCompatActivity() {
19     public var IS_changeded: Int = 0
20     override fun onCreate(savedInstanceState: Bundle?) {
21         super.onCreate(savedInstanceState)
22         setContentView(R.layout.add_note)
23         supportActionBar?.hide()
24
25         window.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SYSTEM_BAR_BACKGROUNDS);
26         window.setStatusBarColor(this.resources.getColor(R.color.back))
27
28         edit_title.onFocusChangeListener = View.OnFocusChangeListener { v,
29             hasFocus ->
30                 if (hasFocus) {
31                     // 此处为得到焦点时的处理内容
32                     save.setVisibility(View.VISIBLE);
33                     IS_changeded = 1
34                 } else {
35                     IS_changeded -= 1
36                     save.setVisibility(View.GONE);
37                     // 此处为失去焦点时的处理内容
38                 }
39             }
40
41         content_edit.onFocusChangeListener = View.OnFocusChangeListener { v,
42             hasFocus ->
43                 if (hasFocus) {
44                     save.setVisibility(View.VISIBLE);
45                     IS_changeded = 1
46                     // 此处为得到焦点时的处理内容
47                 } else {
48                     save.setVisibility(View.GONE);
49                     // 此处为失去焦点时的处理内容
50                 }
51             }
52
53         save.setOnClickListener {
54             IS_INSERT=1
55             IS_changeded=0
56             title_ = edit_title.text.toString()
57             time_ = DateToString.DateToString(Date()).subSequence(0,
58                 10).toString()
59             content_ = content_edit.text.toString()
60             edit_title.clearFocus()
61             content_edit.clearFocus()
62
63             val dbHelper = MyDatabaseHelper(this, "Note.db", 1)
64             val db = dbHelper.writableDatabase
65             val cursor = db.rawQuery("select count(id) from Note", null)
66             cursor.moveToFirst()
67             val number=cursor.getInt(0)
68             val aaa=1
69             db.execSQL("insert into Note (id,title,content,time)
70                 values(?,?,?,?)", arrayOf(number+1,title_,

```

```

66         content_, time_))
67 //         db.execSQL("insert into Note (title,content,time) values(?,?,?)",
arrayOf(title_,
68 //         content_, time_))
69         db.close()
70         dbHelper.close()
71     }
72
73     return_change.setOnClickListener {
74         if(IS_changed==0){
75             finish()
76         } else{
77             AlertDialog.Builder(this).apply {
78                 setMessage("是否保存更改")
79                 setPositiveButton("保存") {
80                     dialog,which->
81
82                         title_ = edit_title.text.toString()
83                         time_ =
DateToString.DateToString_(Date()).subSequence(0, 10).toString()
84                         content_ = content_edit.text.toString()
85                         IS_INSERT=2
86                         finish()
87                 }
88                 setNegativeButton("不保存") {
89                     dialog, which->
90                         finish()
91                 }
92                 show()
93             }
94         }
95     }
96 }
97
98
99 }

```

DateToString.kt

```

1  package com.example.note_two
2
3  import java.text.SimpleDateFormat
4  import java.util.*
5
6  class DateToString {
7      companion object {
8          public fun DateToString_(date: Date):String{
9              val format:String = "yyyy-MM-dd HH:mm:ss";
10             val formatter = SimpleDateFormat(format)
11             return formatter.format(date)
12         }
13     }
14 }

```


MyDatabaseHelper.kt

```
1 package com.example.note_two
2
3 import android.content.Context
4 import android.database.sqlite.SQLiteDatabase
5 import android.database.sqlite.SQLiteOpenHelper
6 import android.widget.Toast
7
8 class MyDatabaseHelper(val context: Context, name: String, version: Int) :
9     SQLiteOpenHelper(context, name, null, version){
10
11     private val createNote = "create table Note("+
12         "id integer primary key ,"+
13         "title text,"+
14         "content text,"+
15         "time text)"
16
17
18     override fun onCreate(p0: SQLiteDatabase?) {
19         p0?.execSQL(createNote)
20         // Toast.makeText(context,"Create table Book
21         // successful!",Toast.LENGTH_SHORT).show()
22     }
23
24     override fun onUpgrade(p0: SQLiteDatabase?, p1: Int, p2: Int) {
25         p0?.execSQL("drop table if exists Note")
26         onCreate(p0)
27     }
28 }
```

Note.kt

```
1 package com.example.note_two
2
3 class Note(var title:String, var content:String, var id_num:Int)
```

NoteAdapter.kt

```
1 package com.example.note_two
2
3 import android.app.AlertDialog
4 import android.content.Intent
5 import android.view.LayoutInflater
6 import android.view.View
7 import android.view.View.OnClickListener
8 import android.view.ViewGroup
9 import android.widget.Filter
10 import android.widget.Filterable
11 import android.widget.TextView
12 import android.widget.Toast
13 import androidx.recyclerview.widget.RecyclerView
14 import java.util.*
15
16 class NoteAdapter(val noteList: MutableList<Note>):
17     RecyclerView.Adapter<NoteAdapter.ViewHolder>(), Filterable{
```

```

17
18 //存放原数据
19 private var mSourceList = mutableListOf<Note>()
20 //存放过滤后的数据
21 private var mFilterList = mutableListOf<Note>()
22
23 private var onItemClickListener: OnItemClickListener? = null
24
25 interface OnItemClickListener {
26     fun onItemClick(view: View?, pos: Int)
27 }
28
29 fun setOnItemClickListener(onItemClickListener: OnItemClickListener?) {
30     this.onItemClickListener = onItemClickListener
31 }
32
33
34 init {
35     mSourceList = noteList
36 }
37
38 inner class ViewHolder(view: View) : RecyclerView.ViewHolder(view){
39     val title_: TextView = view.findViewById(R.id.note_title)
40     val content_: TextView = view.findViewById(R.id.note_content)
41     var id_ :Int = 0
42 }
43
44 override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
ViewHolder {
45     val view =
LayoutInflater.from(parent.context).inflate(R.layout.note_item,parent,false)
46     val viewHolder = ViewHolder(view)
47     viewHolder.itemView.setOnClickListener {
48         val position = viewHolder.adapterPosition
49         val note = noteList[position]
50         var i=0
51         var num=position+1
52         while(i<noteList.size){
53
54             if(noteList[i].title.contains(search_text)||noteList[i].content.contains(searc
h_text)){
55                 num -= 1
56                 if(num==0){
57                     break
58                 }
59                 i += 1
60             }
61             val pos:Int = i
62             id_ = pos
63             val intent=Intent(parent.context,add_change::class.java)
64             intent.putExtra("id_num",pos)
65             intent.putExtra("title",noteList[pos].title)
66
67             intent.putExtra("content",noteList[pos].content.subSequence(13,noteList[pos].c
ontent.length).toString())
68
69             intent.putExtra("time",noteList[pos].content.subSequence(0,10).toString())
70             parent.context.startActivity(intent)
71         }
72     }
73 }

```

```

71         viewHolder.itemView.setOnLongClickListener{
72             val position = viewHolder.adapterPosition
73             val note = noteList[position]
74             AlertDialog.Builder(parent.context).apply {
75                 setTitle("提示")
76                 var content:String = "标题:" + note.title + "\n内容:"
77                 if(note.content.length<=25){
78                     content += note.content.subSequence(13,note.content.length)
79                 } else{
80                     content += note.content.subSequence(13,25)
81                     content += "...
82                 }
83                 content += "\n上次修改时间: "
84                 content += note.content.subSequence(0,10)
85                 setMessage("您确定要删除吗? \n"+content)
86                 setPositiveButton("确定") {dialog, which->
87                     var i=0
88                     var num=position+1
89                     while(i<noteList.size){
90
91                         if(noteList[i].title.contains(search_text)||noteList[i].content.contains(searc
92                         h_text)){
93                             num -= 1
94                             if(num==0){
95                                 break
96                             }
97                         }
98                         i += 1
99                     }
100                     val pos:Int = i
101                     val id_num__ = noteList[pos].id_num
102                     noteList.removeAt(pos)
103                     notifyItemRemoved(pos)
104                     i = 0
105                     while(i<noteList.size){
106                         if(noteList[i].id_num>id_num__){
107                             noteList[i].id_num -= 1
108                         }
109                         i += 1
110                     }
111                     val dbHelper = MyDatabaseHelper(parent.context, "Note.db",
112                     1)
113                     val db = dbHelper.writableDatabase
114                     db.execSQL("delete from Note where id=?",
115                     arrayOf(id_num__))
116                     db.execSQL("update Note set id=id-1 where id>?",
117                     arrayOf(id_num__))
118                     db.close()
119                 }
120                 setNegativeButton("取消") {dialog, which->}
121                 show()
122             }
123             return@setOnLongClickListener true
124         }
125     }
126     return viewHolder
127
128     override fun onBindViewHolder(viewHolder: ViewHolder, position: Int) {

```

```

127         val note_ = mFilterList[position]
128         viewHolder.title_.text = note_.title
129         viewHolder.content_.text = note_.content
130         viewHolder.id_ = note_.id_num
131     }
132
133     override fun getItemCount() = mFilterList.size
134
135
136
137
138     //////////////////////////////////////
139
140
141
142     override fun getFilter(): Filter {
143         return object : Filter() {
144             //执行过滤操作
145             override fun performFiltering(charSequence: CharSequence):
FilterResults {
146                 val charString = charSequence.toString()
147                 //存放已过滤的数据
148                 var theFilterList = if (charString.isEmpty()) {
149                     //没有过滤的内容，则使用源数据
150                     mSourceList
151                 } else {
152                     mSourceList.filter { it.title.contains(charString) ||
it.content.contains(charString)}
153                 }
154                 val filterResults = FilterResults()
155                 filterResults.values = theFilterList
156                 return filterResults
157             }
158             //把过滤后的值返回出来并进行更新
159             override fun publishResults(
160                 charSequence: CharSequence,
161                 filterResults: FilterResults
162             ) {
163                 mFilterList = filterResults.values as MutableList<Note>
164                 notifyDataSetChanged()
165             }
166         }
167     }
168
169
170
171
172
173 }

```

SoftKeyBoardListener.kt

```

1 package com.example.note_two
2
3 import android.app.Activity
4 import android.graphics.Rect
5 import android.view.View
6 import android.view.ViewTreeObserver.OnGlobalLayoutListener
7

```

```

8
9 /**
10  * Created by 唐小糖 on 2017/9/6.
11  * 监听软键盘
12  */
13 class SoftKeyBoardListener(activity: Activity) {
14     private val rootView //activity的根视图
15         : View
16     var rootViewVisibleHeight //纪录根视图的显示高度
17         = 0
18     private var onSoftKeyBoardChangeListener: OnSoftKeyBoardChangeListener? =
19 null
20     private fun setOnSoftKeyBoardChangeListener(onSoftKeyBoardChangeListener:
21 OnSoftKeyBoardChangeListener) {
22         this.onSoftKeyBoardChangeListener = onSoftKeyBoardChangeListener
23     }
24     interface OnSoftKeyBoardChangeListener {
25         fun keyBoardShow(height: Int)
26         fun keyBoardHide(height: Int)
27     }
28     companion object {
29         fun setListener(
30             activity: Activity,
31             onSoftKeyBoardChangeListener: OnSoftKeyBoardChangeListener
32         ) {
33             val softKeyBoardListener = SoftKeyBoardListener(activity)
34
35             softKeyBoardListener.setOnSoftKeyBoardChangeListener(onSoftKeyBoardChangeListen
36 er)
37         }
38     }
39     init {
40         //获取activity的根视图
41         rootView = activity.window.decorView
42
43         //监听视图树中全局布局发生改变或者视图树中的某个视图的可视状态发生改变
44
45         rootView.viewTreeObserver.addOnGlobalLayoutListener(OnGlobalLayoutListener { //
46 获取当前根视图在屏幕上显示的大小
47             val r = Rect()
48             rootView.getWindowVisibleDisplayFrame(r)
49             val visibleHeight = r.height()
50             println(" " + visibleHeight)
51             if (rootViewVisibleHeight == 0) {
52                 rootViewVisibleHeight = visibleHeight
53                 return@OnGlobalLayoutListener
54             }
55
56             //根视图显示高度没有变化，可以看作软键盘显示/隐藏状态没有改变
57             if (rootViewVisibleHeight == visibleHeight) {
58                 return@OnGlobalLayoutListener
59             }
60
61             //根视图显示高度变小超过200，可以看作软键盘显示了
62             if (rootViewVisibleHeight - visibleHeight > 200) {
63                 if (onSoftKeyBoardChangeListener != null) {

```

```

61     onSoftKeyBoardChangeListener!!.keyBoardShow(rootViewVisibleHeight -
    visibleHeight)
62         }
63         rootViewVisibleHeight = visibleHeight
64         return@OnGlobalLayoutListener
65     }
66
67     //根视图显示高度变大超过200, 可以看作软键盘隐藏了
68     if (visibleHeight - rootViewVisibleHeight > 200) {
69         if (onSoftKeyBoardChangeListener != null) {
70             onSoftKeyBoardChangeListener!!.keyBoardHide(visibleHeight -
    rootViewVisibleHeight)
71         }
72         rootViewVisibleHeight = visibleHeight
73         return@OnGlobalLayoutListener
74     }
75 })
76 }
77 }

```

SpacesItemDecoration.kt

```

1  package com.example.note_two
2
3  import android.graphics.Rect
4  import android.view.View
5  import androidx.recyclerview.widget.RecyclerView
6  import androidx.recyclerview.widget.RecyclerView.ItemDecoration
7
8
9  class SpacesItemDecoration(private val space: Int) : ItemDecoration() {
10      override fun getItemOffsets(
11          outRect: Rect, view: View,
12          parent: RecyclerView, state: RecyclerView.State
13      ) {
14          outRect.left = space
15          outRect.right = space
16          outRect.bottom = space
17
18          // Add top margin only for the first item to avoid double space between
    items
19          if (parent.getChildPosition(view) == 0) outRect.top = space
20      }
21  }
22

```

activity_main.xml

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:tools="http://schemas.android.com/tools"
4      android:layout_width="match_parent"
5      android:layout_height="match_parent"
6      android:id="@+id/main_xml"
7
8      android:orientation="vertical"

```

```

9      tools:context=".MainActivity">
10
11      <LinearLayout
12          android:id="@+id/title_activity"
13          android:layout_alignParentTop="true"
14          android:layout_marginTop="5dp"
15          android:orientation="horizontal"
16          android:layout_width="match_parent"
17          android:background="#ffffff"
18          android:layout_height="wrap_content" >
19
20
21
22          <SearchView
23              android:id="@+id/searchView"
24              android:layout_width="0dp"
25              android:layout_weight="5"
26              android:layout_height="wrap_content"
27              android:iconifiedByDefault="false"
28              android:queryBackground="@null"
29              android:submitBackground="@null"
30              android:queryHint="搜索内容" />
31
32
33
34          <RelativeLayout
35              android:id="@+id/menu_activity_container"
36              android:layout_width="0dp"
37              android:layout_height="match_parent"
38              android:layout_weight="1" >
39
40              <Button
41                  android:id="@+id/menu_activity"
42                  android:layout_width="match_parent"
43                  android:layout_height="match_parent"
44                  android:layout_gravity="center"
45                  android:layout_marginStart="12dp"
46                  android:layout_marginTop="12dp"
47                  android:layout_marginEnd="18dp"
48                  android:layout_marginBottom="12dp"
49                  android:background="@drawable/menu" />
50          </RelativeLayout>
51
52      </LinearLayout>
53
54      <androidx.recyclerview.widget.RecyclerView
55          android:id="@+id/recyclerView"
56          android:layout_width="match_parent"
57          android:layout_height="match_parent"
58          android:layout_marginTop="5dp"
59          android:layout_below="@+id/title_activity" />
60
61      <com.google.android.material.floatingactionbutton.FloatingActionButton
62          xmlns:app="http://schemas.android.com/apk/res-auto"
63          android:id="@+id/fab_plus"
64          android:layout_width="wrap_content"
65          android:layout_height="wrap_content"
66          android:layout_alignParentRight="true"
67          android:layout_alignParentBottom="true"
68          android:layout_marginRight="20dp"
69          android:layout_marginBottom="60dp"

```

```

69         android:backgroundTint="#3300ff"
70         android:scaleType="center"
71         android:src="@drawable/plus_2"
72         android:tint="#ffffff"
73         app:pressedTranslationZ="12dp"
74
75     />
76
77 </RelativeLayout>

```

add_change.xml

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent">
5
6      <LinearLayout
7          android:id="@+id/title_change"
8          android:layout_width="match_parent"
9          android:layout_height="wrap_content"
10         android:layout_alignParentTop="true"
11         android:layout_marginTop="15dp"
12         android:orientation="horizontal">
13
14
15         <Button
16             android:id="@+id/return_change"
17             android:layout_width="0dp"
18             android:layout_height="35dp"
19             android:layout_weight="1"
20             android:gravity="center"
21             android:layout_gravity="center"
22             android:layout_marginLeft="15dp"
23             android:background="@drawable/return1" />
24
25         <View
26             android:layout_width="0dp"
27             android:layout_height="match_parent"
28             android:layout_weight="6" />
29
30         <Button
31             android:id="@+id/save"
32             android:layout_width="0dp"
33             android:layout_height="match_parent"
34             android:layout_weight="1"
35             android:layout_marginRight="15dp"
36             android:background="@drawable/save_1"
37             android:visibility="gone"
38             android:gravity="right" />
39
40     </LinearLayout>
41
42
43
44
45     <EditText
46         android:id="@+id/edit_title"
47         android:layout_width="match_parent"

```



```

48         android:layout_height="wrap_content"
49         android:maxLines="1"
50         android:hint="标题"
51         android:textColor="#000000"
52         android:textSize="30sp"
53         android:layout_below="@+id/title_change"
54         android:layout_marginRight="15dp"
55         android:background="@null"
56         android:layout_marginTop="20dp"
57         android:layout_marginLeft="15dp"
58         android:textStyle="bold"/>
59
60     <TextView
61         android:id="@+id/dateText"
62         android:layout_width="match_parent"
63         android:layout_height="wrap_content"
64         android:layout_below="@+id/edit_title"
65         android:layout_marginLeft="15dp"
66         android:layout_marginTop="5dp"
67         android:layout_marginRight="15dp"
68         android:textSize="15sp" />
69
70     <EditText
71         android:id="@+id/content_edit"
72         android:layout_width="match_parent"
73         android:layout_height="match_parent"
74         android:layout_marginLeft="15dp"
75         android:layout_marginRight="15dp"
76         android:layout_below="@+id/dateText"
77         android:background="@null"
78         android:layout_marginTop="10dp"
79         android:gravity="top"
80         android:inputType="textMultiLine"
81         android:textSize="16sp"
82         android:textColor="#000000"/>
83
84
85 </RelativeLayout>

```

add_note.xml

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent"
5      >
6
7      <LinearLayout
8          android:id="@+id/title_change"
9          android:layout_width="match_parent"
10         android:layout_height="wrap_content"
11         android:layout_alignParentTop="true"
12         android:layout_marginTop="15dp"
13         android:orientation="horizontal">
14
15
16         <Button
17             android:id="@+id/return_change"
18             android:layout_width="0dp"

```

```

19         android:layout_height="35dp"
20         android:layout_weight="1"
21         android:gravity="center"
22         android:layout_gravity="center"
23         android:layout_marginLeft="15dp"
24         android:background="@drawable/return1" />
25
26     <View
27         android:layout_width="0dp"
28         android:layout_height="match_parent"
29         android:layout_weight="6" />
30
31     <Button
32         android:id="@+id/save"
33         android:layout_width="0dp"
34         android:layout_height="match_parent"
35         android:layout_weight="1"
36         android:layout_marginRight="15dp"
37         android:background="@drawable/save_1"
38         android:visibility="gone"
39         android:gravity="right" />
40
41 </LinearLayout>
42
43
44
45
46 <EditText
47     android:id="@+id/edit_title"
48     android:layout_width="match_parent"
49     android:layout_height="wrap_content"
50     android:maxLines="1"
51     android:hint="标题"
52     android:textColor="#000000"
53     android:textSize="30sp"
54     android:layout_below="@+id/title_change"
55     android:layout_marginRight="15dp"
56     android:background="@null"
57     android:layout_marginTop="20dp"
58     android:layout_marginLeft="15dp"
59     android:textStyle="bold"/>
60
61 <TextClock
62     android:id="@+id/dateText"
63     android:layout_width="match_parent"
64     android:layout_height="wrap_content"
65     android:layout_below="@+id/edit_title"
66     android:layout_marginLeft="15dp"
67     android:layout_marginTop="5dp"
68     android:layout_marginRight="15dp"
69     android:format12Hour="yyyy-MM-dd HH:MM"
70     android:format24Hour="yyyy-MM-dd HH:MM"
71     android:textSize="15sp" />
72
73 <EditText
74     android:id="@+id/content_edit"
75     android:layout_width="match_parent"
76     android:layout_height="match_parent"
77     android:layout_marginLeft="15dp"
78     android:layout_marginRight="15dp"
79     android:layout_below="@+id/dateText"

```

```
80         android:background="@null"
81         android:layout_marginTop="10dp"
82         android:gravity="top"
83         android:inputType="textMultiLine"
84         android:textSize="16sp"
85         android:textColor="#000000"/>
86
87
88     </RelativeLayout>
```

menu_title.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <menu xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto">
4      <item
5          android:id="@+id/action_crack"
6          android:title="切换视图"
7          app:showAsAction="never"/>
8
9  </menu>
```

note_item.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.cardview.widget.CardView
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:app="http://schemas.android.com/apk/res-auto"
5      android:layout_width="match_parent"
6      android:layout_height="wrap_content"
7      app:cardCornerRadius="15dp"
8      app:cardBackgroundColor="#ffffff"
9      >
10     <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
11         android:orientation="vertical"
12         android:layout_width="match_parent"
13         android:layout_marginLeft="10dp"
14         android:layout_marginRight="10dp"
15
16         android:layout_height="wrap_content">
17
18         <TextView
19             android:id="@+id/note_title"
20             android:layout_width="match_parent"
21             android:layout_height="wrap_content"
22             android:layout_marginTop="13dp"
23             android:textSize="18sp"
24             android:textColor="#000000"
25
26             android:maxLines="1"
27             />
28
29         <TextView
30             android:id="@+id/note_content"
31             android:layout_width="match_parent"
32             android:layout_height="wrap_content"
33             android:layout_marginTop="5dp"
```

```
33         android:maxLines="1"
34         android:textSize="15sp"
35         android:layout_marginBottom="13dp"
36     />
37
38
39     </LinearLayout>
40
41 </androidx.cardview.widget.CardView>
```