

说明文档

1. 软件名称

我的日记本

2. 完成人

彭翊桐 20192131018

3. 完成时间

2021 年 12 月 16 日

4. 内容简介

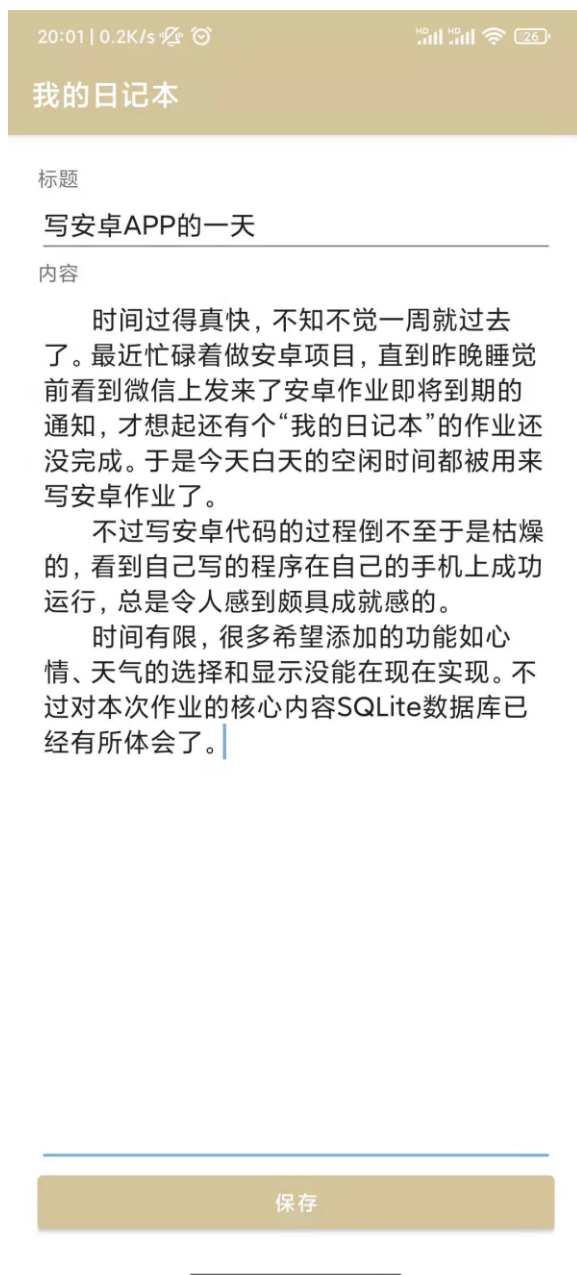
本软件采用 SQLite 技术，实现了日记列表的浏览、日记添加和日记详细内容的查看功能。

5. 界面设计：

主界面：



点击“添加”或“+”进入日记编写界面：



点击“保存”按钮，保存日记，返回主界面，并提示保存成功：



6. 代码设计：

主界面：

```
private lateinit var appBarConfiguration: AppBarConfiguration

private lateinit var binding: ActivityMainBinding

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
```

```

binding = ActivityMainBinding.inflate(layoutInflater)
setContentView(binding.root)

setSupportActionBar(binding.toolbar)

binding.fab.setOnClickListener { view ->
    val intent = Intent(this, AddDiary::class.java)
    startActivity(intent)
}

}

override fun onResume() {
    super.onResume()
    val dbHelper = MyDatabaseHelper(this, "Diary.db", 1)
    val db = dbHelper.readableDatabase
    val titles = arrayListOf<String>()
    val ids = arrayListOf<Int>()
    val cursor = db.query("Diary", null, null, null, null, null, null)
    if (cursor.moveToFirst()) {
        do {
            titles.add(cursor.getString(cursor.getColumnIndex("title")))
            ids.add(cursor.getInt(cursor.getColumnIndex("id")))
        } while (cursor.moveToNext())
    }
    cursor.close()

    val layoutManager = LinearLayoutManager(this)
    recyclerView.layoutManager = layoutManager
    val adapter = DiaryAdapter(titles, ids)
    recyclerView.adapter = adapter
}

override fun onCreateOptionsMenu(menu: Menu): Boolean {
    // Inflate the menu; this adds items to the action bar if it is
    present.
    menuInflater.inflate(R.menu.menu_main, menu)
    return true
}

```

```

override fun onOptionsItemSelected(item: MenuItem): Boolean {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    return when (item.itemId) {
        R.id.action_add -> {
            val intent = Intent(this, AddDiary::class.java)
            startActivity(intent)
            return true
        }
        else -> super.onOptionsItemSelected(item)
    }
}

```

主界面日记列表的适配器:

```

class DiaryAdapter (val diaryList: List<String>, val idList:
List<Int>) : RecyclerView.Adapter<DiaryAdapter.ViewHolder>() {
    inner class ViewHolder (view: View) :
RecyclerView.ViewHolder (view) {
        val diaryTitle: TextView =
view.findViewById(R.id.diary_item_title)
    }

    override fun onCreateViewHolder (parent: ViewGroup, viewType:
Int): ViewHolder {
        val view = LayoutInflater.from (parent.context)
            .inflate (R.layout.diary_item, parent, false)

        /*注册点击事件*/
        val viewHolder = ViewHolder (view)
        viewHolder.itemView.setOnClickListener {
            val position = viewHolder.adapterPosition
            val context: Context = view.context
            val intent = Intent (context, DiaryDetails::class.java)
            intent.putExtra ("ID", idList [position])
            context.startActivity (intent)
        }

        return viewHolder
    }

    override fun onBindViewHolder (holder: ViewHolder, position: Int)
{

```

```

        val diary = diaryList[position]
        holder.diaryTitle.text = diary
    }

    override fun getItemCount() = diaryList.size
}

```

数据库类:

```

class MyDatabaseHelper(val context: Context, name: String,
version: Int): SQLiteOpenHelper(context, name, null, version) {
    private val createDiary = "create table Diary (" +
        "id integer primary key autoincrement," +
        "title text," +
        "date text," +
        "content text)"

    override fun onCreate(db: SQLiteDatabase) {
        db.execSQL(createDiary)
    }

    override fun onUpgrade(p0: SQLiteDatabase?, p1: Int, p2: Int) {

    }
}

```

日记添加:

```

class AddDiary : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_add_diary)

        val dbHelper = MyDatabaseHelper(this, "Diary.db", 1)
        val db = dbHelper.writableDatabase

        saveButton.setOnClickListener {
            val current = LocalDateTime.now()
            val formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd")
            val formatted = current.format(formatter)

            val values = ContentValues().apply {
                put("title", title_edit.text.toString())
                put("date", formatted)
                put("content", content_edit.text.toString())
            }

            db.insert("Diary", null, values)
        }
    }
}

```

```

        Toast.makeText(this, "保存成功", Toast.LENGTH_SHORT).show()
        finish()
    }
}
}

```

日记详细:

```

class DiaryDetails : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_diary_details)

        val id = intent.getIntExtra("ID", 0)
        val title = arrayListOf<String>()
        val date = arrayListOf<String>()
        val content = arrayListOf<String>()
        val dbHelper = MyDatabaseHelper(this, "Diary.db", 1)
        val db = dbHelper.readableDatabase
        val cursor = db.query("Diary", null, "id = ?",
arrayOf(id.toString()), null, null, null)
        if (cursor.moveToFirst()) {
            do {

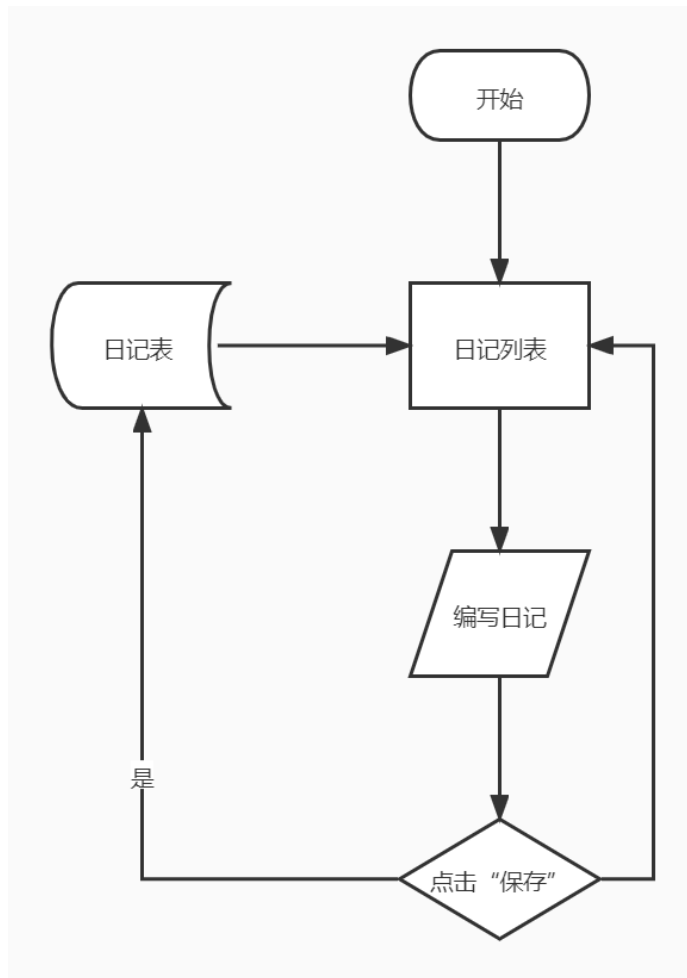
title.add(cursor.getString(cursor.getColumnIndex("title")))

date.add(cursor.getString(cursor.getColumnIndex("date")))

content.add(cursor.getString(cursor.getColumnIndex("content")))
            }while (cursor.moveToNext())
        }
        cursor.close()
        if (!title.isEmpty()) {
            title_detail.text = title[0]
            date_detail.text = date[0]
            content_detail.text = content[0]
        }
    }
}

```

7. 软件操作流程:



8. 难点和解决方案：

- a) 日记列表适配器的编写，虽然书上有实例代码，但是需要理解代码的作用并根据自己的需求进行修改。
- b) SQLite 数据库的使用，有些函数参数较多，第一次使用时较麻烦。

9. 不足之处：

不能进行日记的删除和编辑。

10. 今后的设想：

- a) 添加日记删除、编辑功能
- b) 添加心情、天气选择和显示功能