

# Android 程式設計一期末專案

fragment_home	
2023/01/01	-250元
早餐	-125元
晚餐	-125元
2023/01/02	-200元
午餐	-100元
交通	-100元
	
<a href="#">新增</a>	

指導老師：許子衡老師

班 級：四技資工三甲

姓 名：林庠葳 4A9G0069

王爵堃 4A9G0033

## 1. 摘要

在現今的社會的日常生活當中，做任何事情幾乎脫離不了消費，又因為現今一般人基本上都會使用手機，因此本專案將透過 Android Studio 設計介面，搭配小畫家 3D、Photoshop 等編輯軟體來製作 ICON、使用者 UI 圖像，來開發一套沒有紙筆也能利用手機來記帳的 App。

## 2. 緒論

在這 App 被廣泛使用的時代下，記帳的方式從紙筆有逐漸轉向由電子產品來記錄的趨勢，尤其是人們的生活步調逐年升高，導致外食族的人數一直居高不下，而人一天當中基本上最大開銷就是飲食這部分，因此本專案的記帳 App 除了一般的紀錄支出收入之外，也會加入可選擇日期、金錢的類型。

## 3. 本文

### (1)研究動機

當初得知此課程的期末是要做一個專案時，我們煩惱了很久如何製作一個既不會佔手機空間，又貼切我們日常生活且實用的 App，想到我們兩個人都是從北部離鄉背井下南部唸書的學生，基本上每天的開銷都得由自己掌握，也因為如此我們有了記帳的習慣，讓自己能確實的了解到平時的消費習慣，所以靈機一動就決定做個記帳 App，以消費者的角度來思考一般記帳 App 該如何設計才能讓一般人方便使用的 App。

### (2)相關之課程章節

Week 10 使用 RecyclerView 製作列表

Week 11 製作 icon 與美化介面

Week 12 製作 icon 與美化介面

Week 15 fragment 组件

Week 16 fragment 组件

(3)系統畫面，操作說明

1.進入 APP 後可以看到收入支出的資料、點選新增來記錄收入或支出



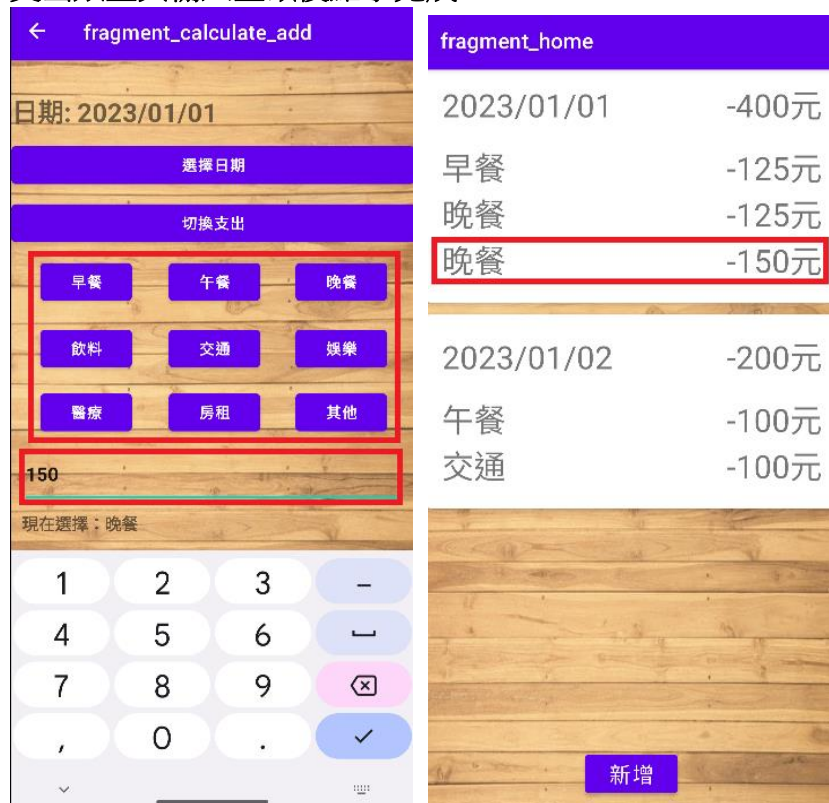
2.先點選切換支出來選擇收入或支出



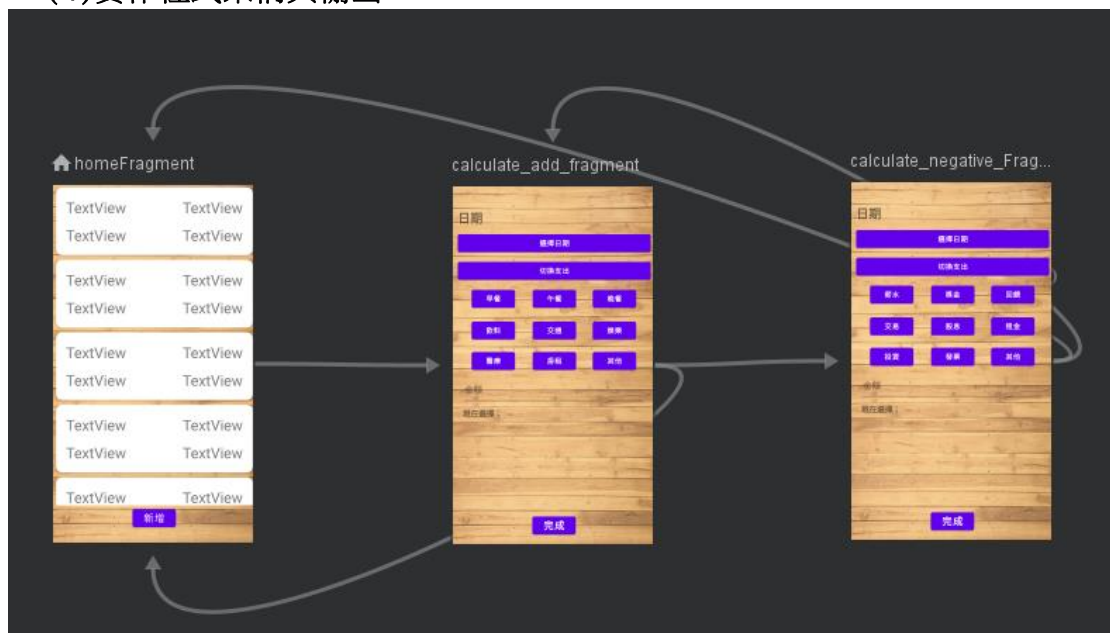
### 3. 點擊選擇日期後選擇要日期



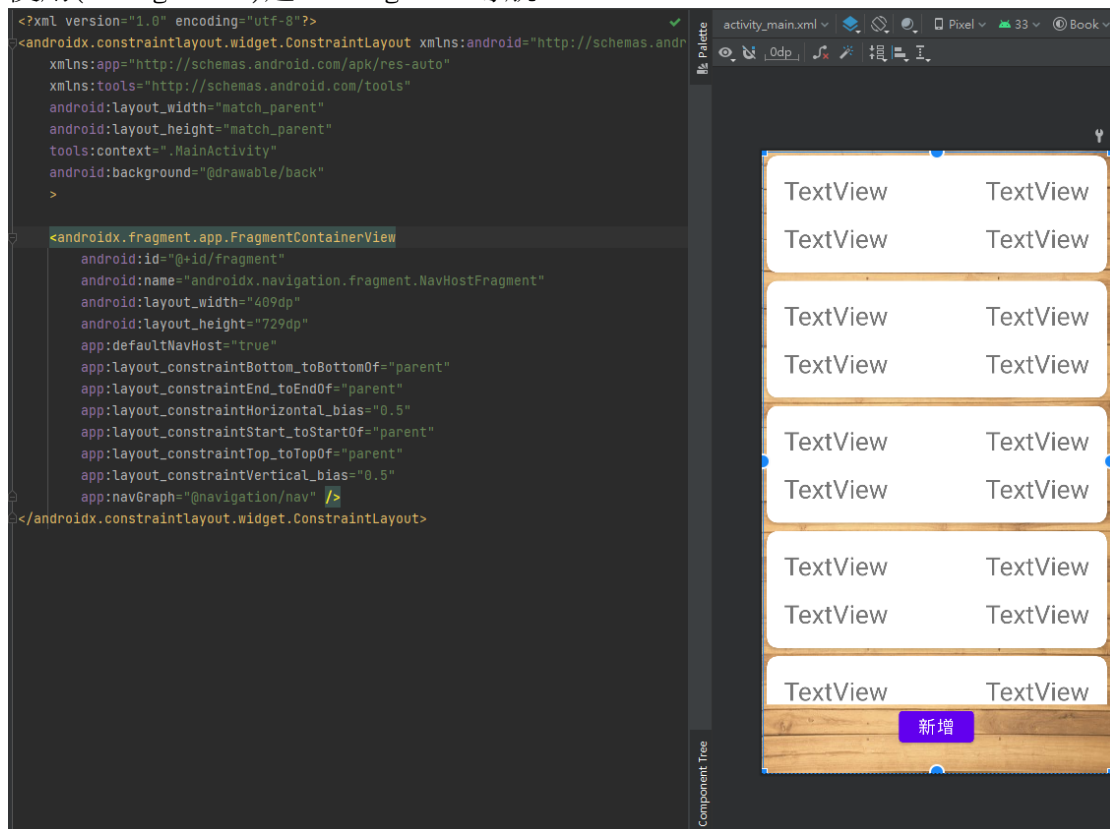
### 4. 選擇支出類型與輸入金額後點擊完成



#### (4)實作程式架構與輸出



使用(navigation)建立 fragment 導航



在 mainactivity 使用 Fragment Container View 建立連結到 homefragment



```

package com.example.book

import ...

class MainActivity : AppCompatActivity() {

    lateinit var binding: ActivityMainBinding
    private lateinit var navController: NavController
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)
        val navHostFragment = supportFragmentManager
            .findFragmentById(R.id.fragment) as NavHostFragment
        navController = navHostFragment.navController

        setupActionBarWithNavController(navController)
    }

    override fun onSupportNavigateUp(): Boolean {
        return navController.navigateUp() || super.onSupportNavigateUp()
    }
}

```

## mainactivity 程式

The screenshot displays the Android Studio environment. On the left, the XML layout file `fragment_home.xml` is open, showing the following structure:

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".homeFragment"
    android:background="@drawable/back">

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="新增"
        android:textSize="20sp"
        app:layout_constraintBottom_toTopOf="@+id/guideline2"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent" />

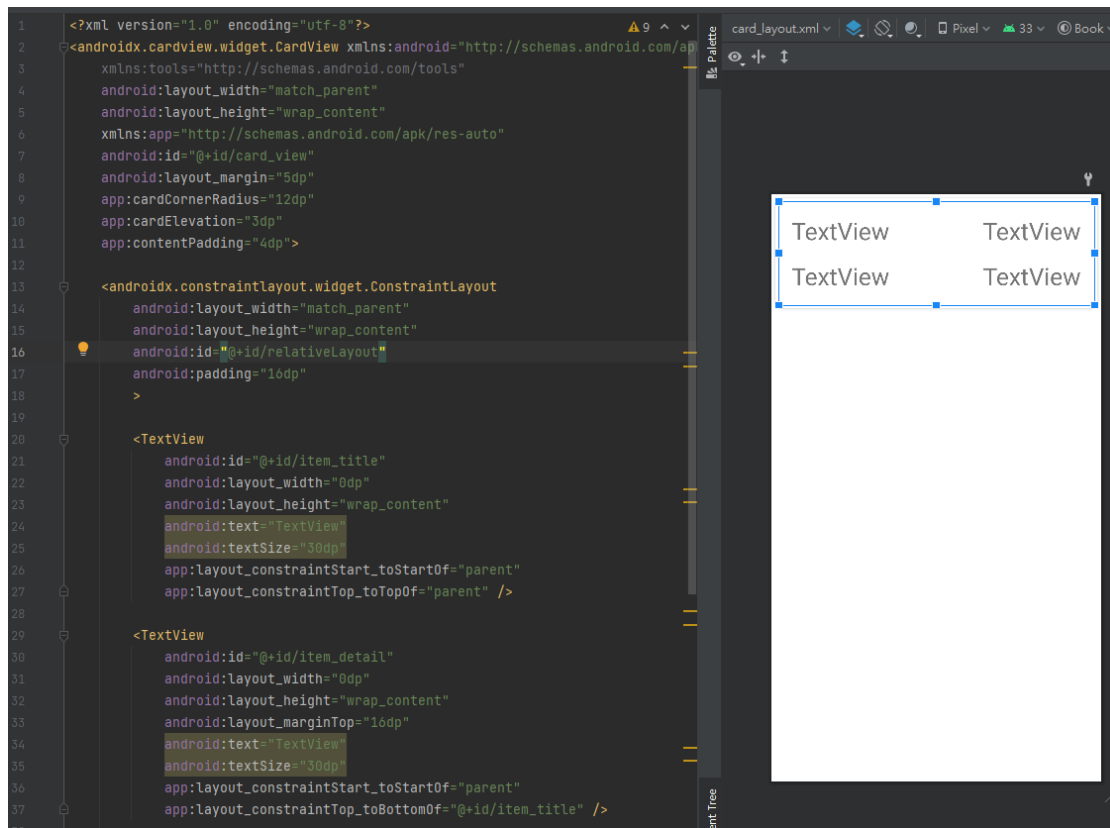
    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        app:layout_constraintGuide_begin="760dp" />

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/bookdata"
        android:layout_width="409dp"
        android:layout_height="650dp"
        android:layout_marginBottom="2dp"
        app:layout_constraintBottom_toTopOf="@+id/button"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:listitem="@layout/card_layout" />

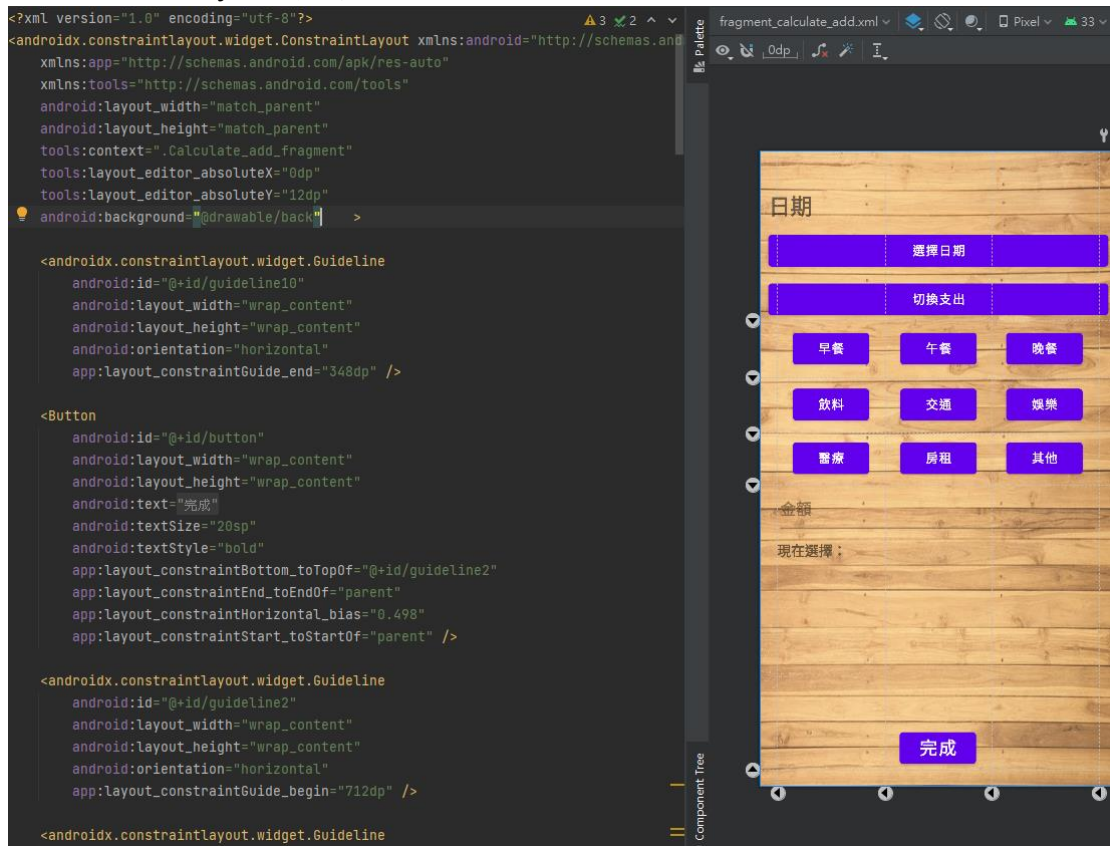
```

On the right, the visual preview of the `fragment_home.xml` is shown. It features a light blue background with a wooden-textured bottom bar. A vertical list of six cards is displayed, each containing two `TextView` elements. A purple button labeled "新增" (Add) is positioned at the bottom right of the list.

homefragment 使用 recyclerview 並且以預設 cardview 建立視窗



## cardview的 layout 設計



記帳支出版面使用 Button 選擇支出類別在使用 EditText 打入金額再用 Button 傳送資料給 homefragment

```

1 package com.example.book
2
3 import androidx.appcompat.app.AppCompatActivity
4
5 class Calculate_add_fragment : Fragment() {
6     private var _binding : FragmentCalculateAddBinding? = null
7     private val binding get() = _binding!!
8     private var nowch = ""
9     private var date = ""
10
11     override fun onCreateView(
12         inflater: LayoutInflater, container: ViewGroup?,
13         savedInstanceState: Bundle?
14     ): View? {
15         _binding = FragmentCalculateAddBinding.inflate(inflater, container, attachToParent: false)
16
17         return binding.root
18     }
19
20     override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
21         super.onViewCreated(view, savedInstanceState)
22
23         binding.button.setOnClickListener { it: View!
24             if (binding.number.text.toString().isEmpty() && nowch != ""){
25                 val action = Calculate_add_fragmentDirections.actionCalculateAddFragmentToHomeFragment( numberData: binding.number.text.toString().toInt()*-1,nowch, date)
26                 Navigation.findNavController(view).navigate(action)
27             }
28             else{
29                 Toast.makeText(activity, text: "請輸入數值且選擇項目", Toast.LENGTH_LONG).show()
30             }
31         }
32
33         binding.buttonSW.setOnClickListener { it: View!
34             findNavController().navigate(R.id.action_calculate_add_fragment_to_calculate_negative_fragment)
35         }
36
37         binding.buttonSalary.setOnClickListener { it: View!
38             binding.textViewch.text = "現在選擇：早餐"
39             nowch = "早餐";
40         }
41
42         binding.buttonGiveBack.setOnClickListener { it: View!
43             binding.textViewch.text = "現在選擇：晚餐"
44             nowch = "晚餐";
45         }
46
47         binding.buttonTrade.setOnClickListener { it: View!
48             binding.textViewch.text = "現在選擇：飲料"
49             nowch = "飲料";
50         }
51
52         binding.buttonCollectRent.setOnClickListener { it: View!
53             binding.textViewch.text = "現在選擇：娛樂"
54             nowch = "娛樂";
55         }
56
57         binding.buttonBonus.setOnClickListener { it: View!
58             binding.textViewch.text = "現在選擇：午餐"
59             nowch = "午餐";
60         }
61
62         binding.buttonDividend.setOnClickListener { it: View!
63             binding.textViewch.text = "現在選擇：交通"
64             nowch = "交通";
65         }
66
67         binding.buttonInvest.setOnClickListener { it: View!
68             binding.textViewch.text = "現在選擇：醫療"
69             nowch = "早餐";
70         }
71
72         binding.buttonOther.setOnClickListener { it: View!
73             binding.textViewch.text = "現在選擇：其他"
74             nowch = "其他";
75         }
76
77         binding.buttonBill.setOnClickListener { it: View!
78             binding.textViewch.text = "現在選擇：房租"
79             nowch = "房租";
80         }
81
82         binding.buttonBill.setOnClickListener { it: View!
83             binding.textViewch.text = "現在選擇：房租"
84             nowch = "房租";
85         }
86     }
87 }

```

25~32 行使用 view Binding

37~45 利用 navigation 傳送資料給 homefragment

46~84 用 Button 設定記帳類別



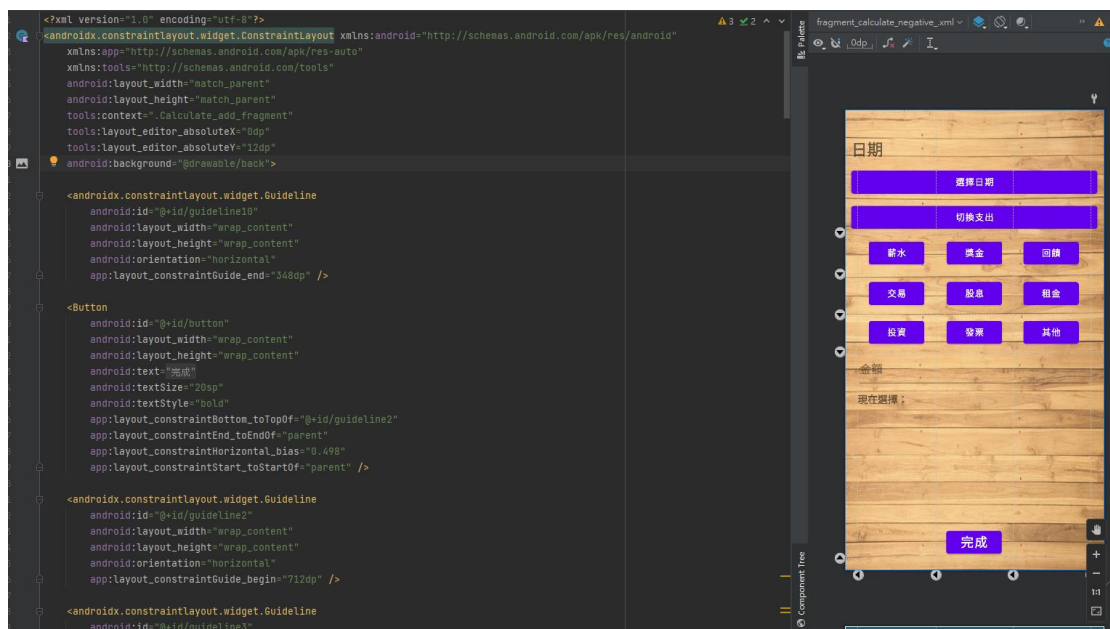
```

binding.buttonSelectDate.setOnClickListener { it: View!
    val datePickerFragment = DatePickerFragment()
    val supportFragmentManager = requireActivity().supportFragmentManager

    supportFragmentManager.setFragmentResultListener(
        requestKey: "REQUEST_KEY", viewLifecycleOwner
    ){
        resultKey, bundle ->
        if (resultKey == "REQUEST_KEY"){15
            date = bundle.getString( key: "SELECTED_DATE").toString()
            binding.textViewDate.text = "日期: " + date
        }
    }
    datePickerFragment.show(supportFragmentManager, tag: "DatePickerFragment")
}

```

85~99 呼叫 DatePickerFragment 類別來監聽選擇時間視窗的 fragment



記帳收入版面與支出版面相同內部程式也大致相同

```

class DatePickerFragment : DialogFragment(), DatePickerDialog.OnDateSetListener {
    private val calendar = Calendar.getInstance()

    override fun onCreateDialog(savedInstanceState: Bundle?): Dialog {
        val year = calendar.get(Calendar.YEAR)
        val month = calendar.get(Calendar.MONTH)
        val day = calendar.get(Calendar.DAY_OF_MONTH)

        return DatePickerDialog(requireActivity(), listener: this, year, month, day)
    }

    override fun onDateSet(view: DatePicker?, year: Int, month: Int, dayOfMonth: Int) {
        calendar.set(Calendar.YEAR, year)
        calendar.set(Calendar.MONTH, month)
        calendar.set(Calendar.DAY_OF_MONTH, dayOfMonth)
        val selectedDate = SimpleDateFormat( pattern: "yyyy/MM/dd", Locale.TAIWAN).format(calendar.time)
        val selectedDateBundle = Bundle()
        selectedDateBundle.putString("SELECTED_DATE", selectedDate)
        setFragmentResult( requestKey: "REQUEST_KEY", selectedDateBundle)
    }
}

```

datepickerdialog 類別 監聽選擇的視窗 選完後存進 selectedDate 並且給上標籤

## (5) 研究過程

在開始做這個 App 之前，我們先以小畫家設計我們的 UI 草圖，並且先擬好各個按鈕的功能與切換 fragment 的按鈕，之後研究 flipclass 上的上課教材與學習大量 youtube 的教學影片，才能順利的在 Activity 與 Fragment 之間做切換並且傳值和如何調用出選擇時間的 fragment，尤其是傳值這個部分耗費我們非常大的心力，因為這跟我們以往熟悉的 C#、C++、C 語言不同，雖然理論都差不多，但實際上操作起來又會遇到蠻多的問題導致要一一排除，不過在排除這些問題後迎來的是自己又進步的成就感。

## (6) 學習心得

當這門課最後一周上完後，我們發現這學期所教的內容，基本上可以順利完成我們所要做的記帳 App，不過可惜的事情就是我們對於資料庫這部分還是一知半解，所以這個記帳的 App 相對的比較不完整，除了這部分比較不完美之外，其他的過程我們都覺得已經達到我們最基本的記帳需求了。

這次也是我們第一次接觸到 kotlin，剛開始覺得這項專案應該難不倒我們，沒想到這跟以往的 C#、C++ 之類的語言不一樣，導致我們處處碰壁，如果以 C# 來做這樣專案的話想必會比用 kotlin 來寫相對快速且不太會出現 bug，不過往好處想，我們至少盡力完成最基本的功能，這樣我們又學習一項屬於自己的技能，而且對開發 Android App 也在這門課中逐漸有興趣，所以即使這門課結束了，我們還會想辦法的把這項專案把它做的更好。

#### 4. 結論

在製作完這項 App 後，我們相信這能為人們的生活上帶來便利，這樣就不用利用紙筆來記錄平時的消費狀況，雖然這個 App 有許多能改善的地方，相信在使用過這個 App 的人都能感同身受，所以這個 App 在未來我們還會改善它所有的功能，並且加入一些新功能上去來讓這個 App 更加的完整與強大。

#### 5. 參考文獻

[DatePickerDialog](#)

[\[Android\] 挑選時間元件 DatePickerDialog 使用方式](#)

[\[Day 28\]Android-DatePickerDialog 的使用](#)

[DatePickerDialog with Fragment - Kotlin](#)

[Navigation Component Passing Data with Safe Args](#)

[How to Navigate Fragments using Navigation Graph](#)

[Recyclerview in Fragment Android Studio Tutorial](#)

[Android Studio App 設置 Activity 背景圖片](#)

[Pexels 圖庫免費下載](#)