國立成功大學　測量及空間資訊學系　113 學年度第 1 學期

行動裝置程式設計　練習六　授課老師：呂學展

I. 作者簡介

A. 姓名：胡瑀真

B. 系級：測量系 116 級

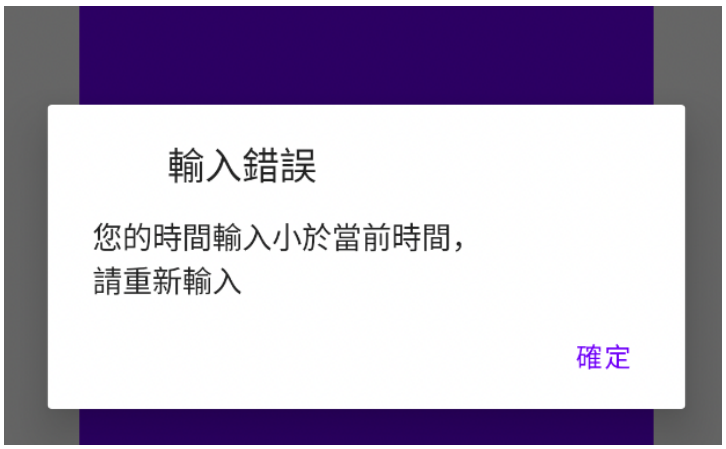
C. 學號：F64126147

D. e-mail：F64126147@gs.ncku.edu.tw

II. 特殊設計

A. 時間的防呆功能

考慮到「購票」在正常情況下，應該只能買「現在」或「未來」的票，因此當選擇的時間為「過去」時，則製作並顯示一個 AlertDialog.Builder ad_time，設定其標題和訊息分別為：輸入錯誤、您的時間輸入小於當前時間，請重新輸入。且使用者只能按下「確定」，按下後，重新出現 DatePickerDialog 或 TimePickerDialog（若使用者是在 DatePickerDialog 時輸入錯誤，則出現 DatePickerDialog，TimePickerDialog 同理）。



圖一、使用者輸入過去的時間時，產生的 AlertDialog.Builder ad_time

偵測選擇的時間是否為過去的方式：運用 Calendar.getInstance().get(Calender.YEAR/MONTH/DAY_OF_MONTH/HOUR_OF_DAY/MINUTE)搭配 if-else 敘述，判斷現在時間是否大於所選擇的時間。判斷時，若以下三種情況中有一種成立，便製作並顯示 AlertDialog.Builder ad_time：現在年份大於選擇年份、年份相同時，現在月份是否大於選擇

月份、年份和月份相同時，現在日期是否大於選擇日期。

小時、分鐘的判斷，與日期的判斷同理。

```
String real_date ="";
@Override
public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {
    Calendar d = Calendar.getInstance();
    int real_year = d.get(Calendar.YEAR);
    int real_month = d.get(Calendar.MONTH);
    int real_dayOfMonth = d.get(Calendar.DAY_OF_MONTH);
    if(real_year>year || (real_year==year && real_month>month) ||
            (real_year==year && real_month==month && real_dayOfMonth > dayOfMonth)){
        ad_time = new AlertDialog.Builder( context: this);
        ad_time.setTitle("輸入錯誤").setMessage("您的時間輸入小於當前時間，\n請重新輸入");
        ad_time.setCancelable(false);
        ad_time.setIcon(android.R.drawable.stat_sys_warning);
        ad_time.setPositiveButton( text: "確定", listener: this);
        ad_time_show=1;
        dpd_show=1;
        tpd_show=0;
        ad_time.show();
    }
    else{
        real_date = real_year+"年"+(real_month+1)+"月"+real_dayOfMonth+"日";
        date = year+"年"+String.format("%02d",(month+1))+"月"+String.format("%02d",dayOfMonth)+"日";
        tpd = new TimePickerDialog( context: this, listener: this,
                d.get(Calendar.HOUR_OF_DAY),d.get(Calendar.MINUTE), is24HourView: true);
        tpd.setCancelable(false);
        dpd_show=0;
        tpd_show=1;
        tpd.show();
    }
```

圖二、函式 onDateSet 的程式碼

```
@Override
public void onTimeSet(TimePicker view, int hourOfDay, int minute) {
    int real_hour = Calendar.getInstance().get(Calendar.HOUR_OF_DAY);
    int real_minute = Calendar.getInstance().get(Calendar.MINUTE);
    if(real_date.equals(date) && (hourOfDay<real_hour || (real_hour == hourOfDay && minute<real_minute))){
        ad_time = new AlertDialog.Builder( context: this);
        ad_time.setTitle("輸入錯誤").setMessage("您的時間輸入小於當前時間，\n請重新輸入");
        ad_time.setCancelable(false);
        ad_time.setIcon(android.R.drawable.stat_sys_warning);
        ad_time.setPositiveButton( text: "確定", listener: this);
        ad_time_show=1;
        tpd_show=1;
        ad_time.show();
    }
    else{
        time = hourOfDay+"時"+String.format("%02d", minute)+"分";
        ad = new AlertDialog.Builder( context: this);
        ad.setTitle("時間資訊");
        ad.setMessage("您選擇的時間為\n"+date+time);
        ad.setCancelable(false);
        ad.setIcon(android.R.drawable.btn_star);
        ad.setPositiveButton( text: "確定", listener: this).setNegativeButton( text: "取消", listener: this);
        ad_show=1;
        ad.show();
    }
}
```

圖三、函式 onTimeSet 的程式碼

B. 票種與張數選擇與顯示

使用監聽器（CompoundButton.OnCheckedChangeListener、View.OnClickListener）以完成票種和張數的選擇和顯示功能。而版面配置如圖一。

圖四、版面配置

票種共有六種，分別為原價票、身障票、學生票、團體票和優待票，為達成「使用者按下 CheckButtom 後才可以在 EditText 中輸入票數」的功能，為每個票種撰寫兩個 if 函式，判斷全票的 CheckButtom 是否有被點選。

以原價票為例：若被點選，則開啟全票對應的 EditText，其允許聚焦、允許觸摸模式聚焦和主動請求焦點的功能（原本預設所有票種的 EditText 都是不可編輯的，使用程式碼 android:focusable="false"）。

若不被點選，則清空對應的 EditText，以方便使用者在取消勾選後，一併取消原本填寫的票數。之後再次關閉 EditText 的聚焦狀態，使 EditText

再次回到不可編輯，直到下次 CheckButttom 被勾選。

```
@Override
public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
    CompoundButton regular = findViewById(R.id.regular);
    CompoundButton student = findViewById(R.id.student);
    CompoundButton group = findViewById(R.id.group);
    CompoundButton challenge = findViewById(R.id.challenge);
    CompoundButton invite = findViewById(R.id.invite);
    EditText ed_regular = findViewById(R.id.ed_regular);
    EditText ed_student = findViewById(R.id.ed_student);
    EditText ed_group = findViewById(R.id.ed_group);
    EditText ed_challenge = findViewById(R.id.ed_challenge);
    EditText ed_invite = findViewById(R.id.ed_invite);

    if(regular.isChecked()){
        ed_regular.setFocusableInTouchMode(true);
        ed_regular.setFocusable(true);
        ed_regular.requestFocus();
    }
    if(regular.isChecked()==false){
        ed_regular.setText("");
        ed_regular.setFocusableInTouchMode(false);
        ed_regular.setFocusable(false);
    }
```

圖五、函式 OnCheckedChange 的部分程式碼

當按下「確定」鍵，以確定購票後，則宣告一變數 i 為 0、Arraylist ticket 和 number，並且用接連使用 if 函式判斷各個票種對應的 EditText 是否不為空。若不為空，則增加該票種進入 ArrayList ticket 中，並將該票種的數量加入 Arraylist number 中，並使 i 增加 1 以利後續的判斷。

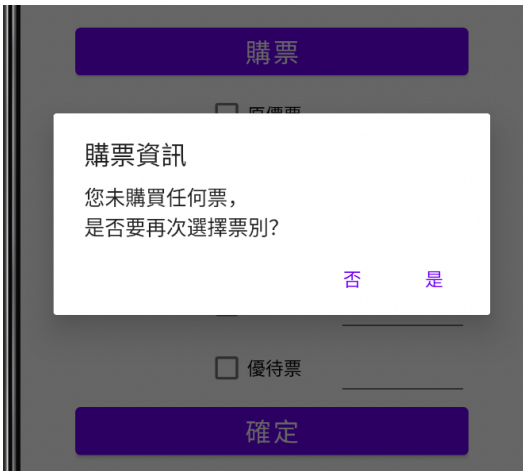
```
@Override
public void onClick(View view) {
    if(view.getId() == R.id.buy){
        Calendar c = Calendar.getInstance();
        dpd = new DatePickerDialog( context: this, listener: this,
                c.get(Calendar.YEAR),c.get(Calendar.MONTH),c.get(Calendar.DAY_OF_MONTH));
        dpd.setCancelable(false);
        tpd_show=0;
        dpd_show=1;
        dpd.show();
    }
    if(view.getId() == R.id.confirm){
        i=0;
        EditText ed_regular = findViewById(R.id.ed_regular);
        EditText ed_student = findViewById(R.id.ed_student);
        EditText ed_group = findViewById(R.id.ed_group);
        EditText ed_challenge = findViewById(R.id.ed_challenge);
        EditText ed_invite = findViewById(R.id.ed_invite);

        ArrayList<String> ticket = new ArrayList<>();
        ArrayList<String> number = new ArrayList<>();

        if(!ed_regular.getText().toString().isEmpty()){
            ticket.add("原價票");
            number.add(ed_regular.getText().toString());
            i++;
        }
```

圖六、函式 onClick，按下「確定」的部分程式碼

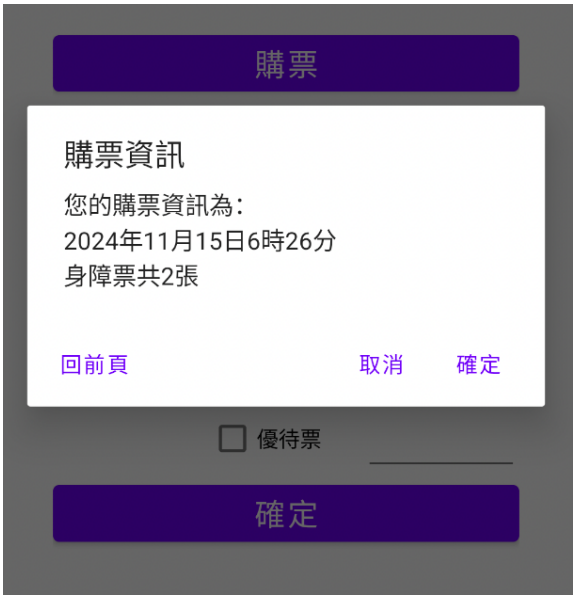
製作一個 AlertDialog.Builder ad_final，以顯示最後的購買結果，並偵測 i 是否為 0。若為 0，則代表沒有任何票種被購買，故 ad_final 的訊息顯示「您未購買任何票，是否要再次選擇票別？」。若使用者在此時點選「是」，則畫面回到票種與數量的選擇；若使用者點選「否」，則浮出「取消購票」的對話框並使程式回到最開始的狀態。



圖七、i 為 0 時，按下「確定」鍵產生的 AlertDialog.Builder

若 i 不為 0，則代表有票種被購買，利用剛剛製作的 Arraylist ticket 和 number 搭配 i 以及 for 迴圈，依序列用 ad_final 印出購買的詳細內容。

另外，i 不為 0 時的 ad_final 相較於 i 等於 0 時的版本，新增「回前頁」的選項，若使用者點選「回前頁」，則可繼續更動表種與數量。而當使用者點選「確定」，則畫面回到程式最開始的狀態，並浮出「購票成功」的對話框；當點選「取消」，則浮出「取消購票」的對話框並使程式回到最開始的狀態。



圖八、i 不為 0 時，按下「確定」鍵產生的 AlertDialog.Builder

```
    ad_final = new AlertDialog.Builder( context: this);
    ad_final.setTitle("購票資訊").setCancelable(false);
    if (i == 0) {
        ad_final.setMessage("您未購買任何票，\n是否要再次選擇票別？");
        ad_final.setPositiveButton( text: "是", listener: this).setNegativeButton( text: "否", listener: this);
    }
    else if(i != 0){
        String answer1 = "您的購票資訊為：\n"+date+time+"\n";
        String answer2 = "";
        for(int j=0;j<i;j++){
            answer2 +=ticket.get(j)+"共"+number.get(j)+"張\n";
        }
        ad_final.setMessage(answer1+answer2);
        ad_final.setPositiveButton( text: "確定", listener: this).setNegativeButton( text: "取消", listener: this)
                .setNeutralButton( text: "回前頁", listener: this);
    }
    ad_final_show=1;
    ad_final.show();
}
```

圖九、函式 onClick，按下「確定」的部分程式碼

C. 團體票的防呆功能

預設團體票至少需購買 11 張，故撰寫防呆設計：在函式 OnEditorAction 中，若第一層 if-else 敘述判斷 EditText 不為空，則進入第二層 if-else 敘述，判斷 EditText 轉型成整數後，是否不小於 11。

於 11，則清空團體票票數輸入匡，並顯示一個 AlertDialog.Builder ad_group，出現「團體票最少需購買 11 張」的提示，且使用者只能按下確定鍵，按下後，畫面回到票種與數量的選擇。

```
@Override
public boolean onEditorAction(TextView v, int actionId, KeyEvent event) {
    EditText ed_group = findViewById(R.id.ed_group);
    if(!ed_group.getText().toString().isEmpty()){
        if(Integer.parseInt(ed_group.getText().toString())<11){
            ed_group.setText("");
            ad_group = new AlertDialog.Builder( context: this);
            ad_group.setTitle("輸入錯誤").setMessage("團體票最少需購買11張").setCancelable(false);
            ad_group.setIcon(android.R.drawable.sym_action_email)
                    .setPositiveButton( text: "確定", listener: this);
            ad.setCancelable(false);
            ad_group_show=1;
            ad_group.show();
        }
    }
    return false;
}
```

圖十、函式 OnEditorAction 中的團體票防呆功能

D. 監聽器 DialogInterface.OnClickListener 中的各種 Dialog 判斷

為了偵測哪一個 Dialog 正在顯示、不同的 Dialog 在 OnClickListener 中有不同的應對方式，我為各種 Dialog 宣告各自對應的整數變數，以代表各種 Dialog 是否正在顯示，若正在顯示，則設為 1，反之與預設值則為

0。

每當 Dialog 顯示，例如：ad.show()，則 ad_show 便令為 1。

```
int i=0,ad_show=0,ad_final_show=0,ad_group_show=0,
        ad_time_show=0,tpd_show=0,dpd_show=0;
```

圖十一、各 Dialog 對應的整數變數

以 ad_show 為例：當 ad 顯示時，ad_show 為 1，又當使用者按下該 Dialog 中的按鍵，則 onClick 函式運用 if 敘述，判斷是哪一個變數為 1。在判斷出是 ad_show 為 1 後，再做該 Dialog 相應的功能判斷。

```
@Override
public void onClick(DialogInterface dialog, int which) {
    Toast t = Toast.makeText( context: this, text: "取消購票",Toast.LENGTH_SHORT);
    Snackbar s = Snackbar.make(findViewById(R.id.root), text: "時間選擇成功",Snackbar.LENGTH_SHORT);
    Snackbar s2 = Snackbar.make(findViewById(R.id.root), text: "購票成功",Snackbar.LENGTH_SHORT);
    LinearLayout tk = findViewById(R.id.tk);
    Button confirm = findViewById(R.id.confirm);

    if(ad_time_show==1){
        if(which == DialogInterface.BUTTON_POSITIVE){
            if(dpd_show==1){
                dpd.show();
            }
            if(tpd_show==1){
                tpd.show();
            }
        }
        tpd_show=0;
        dpd_show=0;
        ad_time_show=0;
    }
    if(ad_show==1){
        if(which == DialogInterface.BUTTON_NEGATIVE){
            t.show();
        }
        if(which == DialogInterface.BUTTON_POSITIVE){
            s.show();
            tk.setVisibility(View.VISIBLE);
            confirm.setVisibility(View.VISIBLE);
        }
        ad_show=0;
    }
```

圖十一、監聽器 DialogInterface.OnClickListener 對應的函式部分程式碼