

鼠婦載具

組員：

資管三 S0961001 陳常葳

資管三 S0961031 方禹心

指導教授：陳巧旻 老師

目錄

壹、系統需求	5
貳、需求塑模	5
一、需求描述	5
(一)、註冊	5
(二)、登入	5
(三)、查看使用者發票	5
(四)、新增發票	5
(五)、發票對獎	6
(六)、角色飼養機制	6
(七)、發票開獎號碼資訊	6
二、結構圖	6
參、資料塑模	6
一、實體關聯圖(ERD)	6
二、正規化	7
(一)、使用者-使用者發票	7
(二)、使用者發票-開獎發票	7
三、關聯表資料字典	7
(一)、資料字典(使用者資料)	7
(二)、資料字典(使用者發票)	8
(三)、資料字典(開獎發票資料)	9
肆、使用者介面塑模	10
一、介面架構圖	10
(一)、階層式介面架構(系統功能表)	10
二、介面藍圖和功能表	10
(一)、註冊	10
(二)、登入	13
(三)、首頁	20
(四)、使用者發票資訊	28
(五)、開獎發票資訊	32
伍、軟硬體環境設計及開發工具	35
一、作業系統	35
二、系統架構	35
三、開發工具	36

表目錄

表 1 資料字典(使用者資料)	7
表 2 資料字典(使用者發票)	8
表 3 資料字典(開獎發票資料)	9

圖目錄

圖 1.....	6
圖 2.....	7
圖 3.....	7
圖 4.....	7
圖 5.....	10
圖 6.....	10
圖 7.....	14
圖 8.....	20
圖 9.....	28
圖 10.....	32
圖 11.....	36

壹、系統需求

系統開發背景

近年來發票載具逐漸普及，不僅響應環保節省紙張成本，也增加生活的機能便利性；由於小組成員於大學活動閒暇期間曾飼養過節肢動物「鼠婦」，經過多次討論後，一致認為以鼠婦作為主題，並以現今發票載具之功能完成應用程式開發。本小組成員全權負責專案的目標、限制、需求、範圍界定、系統分析與設計、程式設計、網路、資料庫管理及規劃；鼠婦載具之目標與限制如下：

系統目標與限制

1. 應用程式須包含三個介面：首頁、讀取持有發票資料之介面、與開獎發票資訊介面。
2. 此應用程式必須提供資料讀出及寫入功能。
3. 使用者須首次使用時須註冊帳號，才可使用該應用程式之所有功能。

貳、需求塑模

一、需求描述

(一)、註冊

鼠婦載具之註冊部分需求描述如下：

1. 使用者於首次使用時須用使用者名稱、email、密碼、手機號碼註冊。

(二)、登入

鼠婦載具之登入部分需求描述如下：

2. 使用者於註冊後，每次使用軟體時皆須以 email 及密碼登入。

(三)、查看使用者發票

鼠婦載具之查看發票部分需求描述如下：

1. 使用者於查看發票介面可察看自己擁有之發票資訊，即發票號碼、發票檔期、中獎金額及兌獎狀態皆可查看

(四)、新增發票

鼠婦載具之新增發票部分需求描述如下：

1. 使用者成功登入系統後，可於首頁以發票號碼與發票年月份來新增自己手邊擁有之發票。

(五)、發票對獎

鼠婦載具之發票對獎部分需求描述如下：

1. 使用者於查看使用者發票介面可點選對獎按鈕，點選可將更新自己擁有之發票狀態及金額，以查看是否中獎。

(六)、角色飼養機制

鼠婦載具之角色飼養機制部分需求描述如下：

1. 每新增 3 張發票升級一次鼠婦樣貌
2. 從「寶寶」角色開始，依序升級為「熊貓王鼠婦」、「魔藥西瓜鼠婦」、「檸檬藍鼠婦」等角色
3. 新增到第 9 張發票就從此為「檸檬藍鼠婦」

(七)、發票開獎號碼資訊

鼠婦載具之發票開獎號碼資訊部分需求描述如下：

1. 於開獎發票資訊分頁將顯示開獎發票之完整號碼及獎項。

二、結構圖

根據完整資料流程圖(DFD)，鼠婦載具之結構圖，如圖 1 所示。

註冊(A)	登入(B)	首頁(C)	使用者發票資訊(D)	開獎發票資訊(E)
註冊(A1)	登入(B1)	新增發票(C1)	查看使用者發票(D1)	開獎發票資訊(E1)
		角色飼養機制(C2)	發票兌獎(D2)	

圖 1 結構圖(鼠婦載具)

參、資料塑模

一、實體關聯圖(ERD)

從鼠婦載具之需求分析結果(藍圖、資料詞彙)，繪製鼠婦載具之實體關聯圖(ERD)，如圖 2 所示。

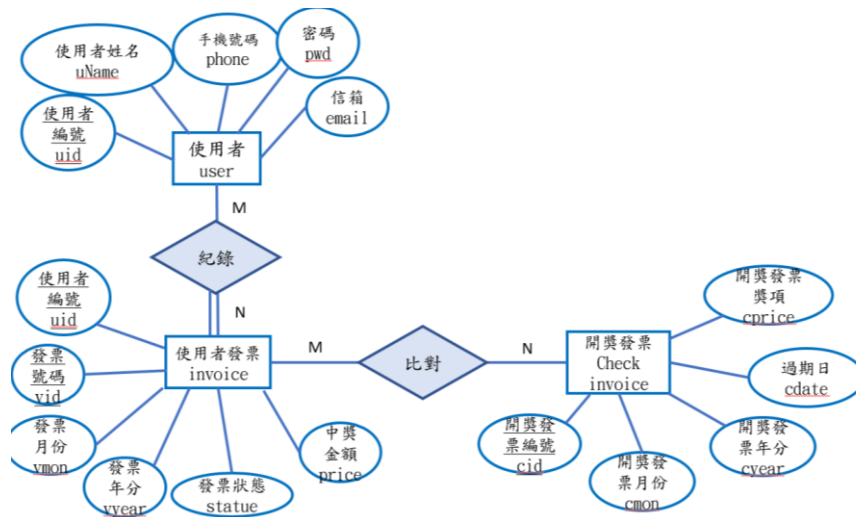


圖 2 ERD 圖(鼠婦載具)

二、正規化

(一)、使用者-使用者發票



圖 3 實體關聯圖(使用者-使用者發票)

(二)、使用者發票－開獎發票

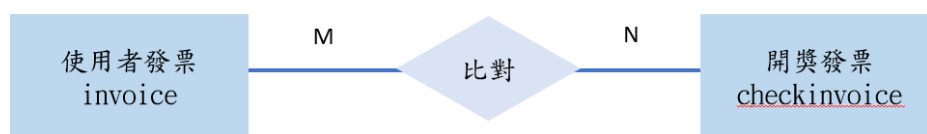


圖 4 實體關聯圖(使用者發票－開獎發票)

三、關聯表資料字典

(一)、資料字典(使用者資料)

依據完成正規化之實體關聯圖，完成關聯表資料字典(商品資料)，如表 1 所示。

表 1 資料字典(使用者資料)

資料元素名稱	欄位名稱	欄位型態	欄位長度	欄位規格/限制	欄位範例	輸入	主鍵	外鍵
使用者編號	uid	varchar	30	英數亂碼	S3j6ji09QZdk29fhlwo10fje9k4		◆	
使用者名稱	uName	varchar	10		張鼠婦	U		
手機號碼	phone	int	10	必須 10 碼	0987654321	U		
信箱	email							
密碼	pwd	varchar	30	最少 6 碼	Abc135	U		

(二)、資料字典(使用者發票)

依據完成正規化之實體關聯圖，完成關聯表資料字典(顧客資料)，如表 2 所示。

表 2 資料字典(使用者發票)

資料元素名稱	欄位名稱	欄位型態	欄位長度	欄位規格/限制	欄位範例	輸入	主鍵	外鍵
使用者編號	uid	varchar	30	英數亂碼	S3j6ji09QZdk29fhlwo10fje9k4			◆
發票編號	vid	varchar	8	必須為 8 碼	12345678	U	◆	
發票月份	vmon	int	2	1、3、5、7、9、11	1	U		
發票年份	vyear	year	4	必須為西元年	2023			
發票狀態	statue	varchar	20	expired/ not yet/ no award/winn ing	expired			
中獎金額	price	varchar	5	1000w/200w /20w/4w/1w /4000/1000 /200/0	200			

(三)、資料字典(開獎發票資料)

依據完成正規化之實體關聯圖，完成關聯表資料字典(使用者資料)，如表 3 所示。

表 3 資料字典(開獎發票資料)

資料元素名稱	欄位名稱	欄位型態	欄位長度	欄位規格/限制	欄位範例	輸入	主鍵	外鍵
開獎發票編號	cid	Varchar	8	必須為 8 碼	09876543	U	◆	
開獎發票月份	cmon	int	2	1、3、5、7、 9、11	3	U		
開獎發票年份	cyear	year	4	必須為西元年	2023			
開獎發票獎項	cprice	Varchar	20	special award/super award/ first award/	first award	U		
過期日	cdate	Date		YYYY/MM/DD	2023/05/05	U		

肆、使用者介面塑模

一、介面架構圖

(一)、階層式介面架構(系統功能表)

依據資料流程圖(DFD)分解之結果，此系統之階層式介面架構(系統功能表)，如圖 5 所示。

註冊(A)	登入(B)	首頁(C)	使用者發票資訊(D)	開獎發票資訊(E)
註冊(A1)	登入(B1)	新增發票(C1)	查看使用者發票(D1)	開獎發票資訊(E1)
		角色飼養機制(C2)	發票兌獎(D2)	

圖 5 階層式介面架構(系統功能表)- 鼠婦載具

二、介面藍圖和功能表

(一)、註冊

介面藍圖(註冊) (activity_register.xml)包含使用者資料欄位，介面藍圖(註冊)如圖 6 所示。

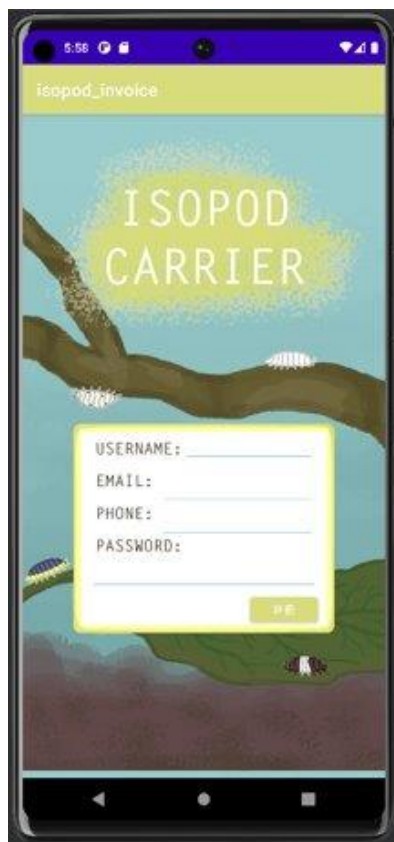


圖 6 介面藍圖(註冊)(activity_register.xml)

```

package com.example.isopod_invoice

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.util.Log
import android.view.MotionEvent
import android.view.View
import android.view.inputmethod.InputMethodManager
import androidx.appcompat.app.AlertDialog
import com.google.firebase.auth.FirebaseAuth
import com.google.firebase.auth.ktx.auth
import com.google.firebase.ktx.Firebase
import com.google.firebase.auth.FirebaseUser

import com.example.isopod_invoice.databinding.ActivityRegisterBinding //import 我們自己的
databinding
import okhttp3.*
import java.io.IOException

class Register : AppCompatActivity() {

    private lateinit var binding: ActivityRegisterBinding
    private lateinit var auth: FirebaseAuth

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityRegisterBinding.inflate(layoutInflater)
        setContentView(binding.root)

        // <-----點桌面鍵盤消失----->S>
        // 獲取最外層的 View
        val rootView: View = findViewById(android.R.id.content)

        // 設置畫面點擊事件監聽器
        rootView.setOnTouchListener { v, event ->
            // 判斷點擊事件的類型
            if (event.action == MotionEvent.ACTION_DOWN) {
                // 獲取輸入法管理器
                val inputMethodManager =
                    getSystemService(INPUT_METHOD_SERVICE) as InputMethodManager

```

```

        // 隱藏鍵盤
        inputMethodManager.hideSoftInputFromWindow(v.windowToken, 0)
    }
    false
}
//<-----點桌面鍵盤消失-----E>

// Step 1: 先初始化 binding 的變數與 FirebaseAuth 變數
auth = FirebaseAuth

//Step 2: 點下 signUp 後需要取得 editview 的 email 與 password 兩個欄位的值
binding.signUp.setOnClickListener {

    //step 2: 取得 email and password
    val email = binding.email.text.toString()
    val password = binding.pwd.text.toString()
    val uName = binding.name.text.toString()
    val phone = binding.phone.text.toString()

    //step 3: auth 內有一個 create user using email and password 的 function
    auth.createUserWithEmailAndPassword(email, password) //把變數 email password 放進去
        .addOnCompleteListener {
            //Step 3:在這裡去判斷傳入的值是否有成功，所以要寫一個判斷式
            if (it.isSuccessful)
            {
                val user = it.result?.user
                val uid = user?.uid

                //新增進去 phpmyadmin
                val clientPost = OkHttpClient()
                val requestBody = FormBody.Builder()
                    .add("uid", "${uid}")
                    .add("uName", "${uName}")
                    .add("phone", "${phone}")
                    .add("email", "${email}")
                    .add("pwd", "${password}")
                    .build()
                val requestPost = Request.Builder()
                    .url("https://fangbibi.lionfree.net/app/userinsert.php")
                    .post(requestBody)
                    .build()
            }
        }
    }
}

```

```

        clientPost.newCall(requestPost).enqueue(object : Callback
        {
            override fun onFailure(call: Call, e: IOException)
            {
                e.printStackTrace()
            }

            override fun onResponse(call: Call, response: Response)
            {
                val responseBody = response.body?.string()
                // Handle the response body
                println("POST-----${responseBody}")
            }
        })

        Log.d("Test", "成功註冊")
        finish()//註冊成功就關閉這個頁面
    }
    else{
        Log.w("Test", "註冊失敗", it.exception)
        println(it.exception)//it.exception 是把錯誤原因記下來
        showMessage("註冊會員失敗")//去呼叫 showMessage 來顯示註冊失敗
    }
}

//Step 3:為了讓使用者知道註冊失敗，所以要寫一個"副程式"showMessage 給使用者知道自己註冊失
敗的訊息
private fun showMessage(message: String) {
    val alertDialog: AlertDialog.Builder = AlertDialog.Builder(this)
    alertDialog.setMessage(message)
    alertDialog.setPositiveButton("確定") { dialog, which -> }
    alertDialog.show()
}
}

```

(二)、登入

介面藍圖(登入)(activity_main.xml)包含使用者資料欄位，介面藍圖(登入)如圖 7 所示。

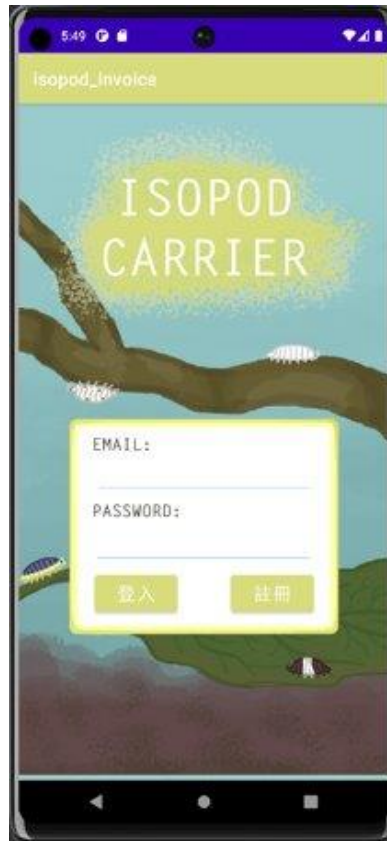


圖 7 介面藍圖(登入)(activity_main.xml)

```
package com.example.isopod_invoice

//import com.google.firebase.auth.FirebaseUser //本次新增
import android.content.Intent
import android.os.Bundle
import android.text.method.ScrollingMovementMethod
import android.view.MotionEvent
import android.view.View
import android.view.inputmethod.InputMethodManager
import android.widget.TextView
import android.widget.Toast
import androidx.appcompat.app.AlertDialog
import androidx.appcompat.app.AppCompatActivity
import com.example.isopod_invoice.databinding.ActivityMainBinding
import com.facebook.*
import com.facebook.appevents.AppEventsLogger
import com.facebook.login.LoginResult
import com.facebook.login.widget.LoginButton
import com.google.firebase.auth.FirebaseAuthProvider
```

```

import com.google.firebase.auth.FirebaseAuth
import com.google.firebase.auth.FirebaseUser
import com.google.firebase.auth.ktx.auth
import com.google.firebase.ktx.Firebase
import com.google.firebase.ktx.initialize
import okhttp3.*
import okhttp3.HttpUrl.Companion.toHttpUrlOrNull
import org.json.JSONObject
import java.io.IOException

class MainActivity : AppCompatActivity() {
    //Step 1: 初始化 FirebaseAuth
    private lateinit var auth: FirebaseAuth
    //Step 1: 先從註冊這個功能開始寫->先初始化一個資料鏈結的部分
    private lateinit var binding: ActivityMainBinding
    private lateinit var callbackManager: CallbackManager
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        FacebookSdk.sdkInitialize(applicationContext)
        AppEventsLogger.activateApp(application)
        setContentView(R.layout.activity_main)

        //<-----點桌面鍵盤消失-----S>
        // 獲取最外層的 View
        val rootView: View = findViewById(android.R.id.content)

        // 設置畫面點擊事件監聽器
        rootView.setOnTouchListener { v, event ->
            // 判斷點擊事件的類型
            if (event.action == MotionEvent.ACTION_DOWN) {
                // 獲取輸入法管理器
                val inputMethodManager =
                    getSystemService(INPUT_METHOD_SERVICE) as InputMethodManager
                // 隱藏鍵盤
                inputMethodManager.hideSoftInputFromWindow(v.windowToken, 0)
            }
            false
        }
        //<-----點桌面鍵盤消失-----E>
    }
}

```

```

/** Step 2: 去設定我們要去啟動的 RegisterActivity **
//Step 2-1: 初始化 FirebaseAuth->初始化 auth
auth = Firebase.auth
Firebase.initialize(this)
auth = FirebaseAuth.getInstance()
callbackManager = CallbackManager.Factory.create()

//Step 2-2: 先從註冊這個功能開始寫->初始化 binding 要鏈結的 layout
binding = ActivityMainBinding.inflate(layoutInflater)
setContentView(binding.root)

//Step 2-3: 先從註冊這個功能開始寫->先宣告要在"signUp" 這個 button 內寫監聽程式，
binding.signUp.setOnClickListener {

    // Step 2-4: 宣告一個 intent 變數，使用 Intent 去開啟一個我們自己寫好的
RegisterActivity
    var intentRegister = Intent(this, Register::class.java)
    // Step 2-5: start 這個 Activity
    startActivity(intentRegister)
}

//Step 3-1: 設計登入的 button
binding.signIn.setOnClickListener {
    if (binding.email.text.toString().isEmpty()) {
        showMessage("請輸入帳號")
    } else if (binding.pwd.text.toString().isEmpty()) {
        showMessage("請輸入密碼")
    } else {
        signIn()
    }
}
}

//Step 3-2: 設計登入的 button->寫一個登入認證失敗的判斷式
private fun signIn() {
    val email = binding.email.text.toString()
    val password = binding.pwd.text.toString()

    auth.signInWithEmailAndPassword(email, password)
        .addOnCompleteListener {
            if (it.isSuccessful) {

```



```

        //Log.d("signInWithEmail:success")
        println("-----signInWithEmail:success-----")
        val user = auth.currentUser
        updateUI(user)
        val auth1 = FirebaseAuth.getInstance()

// 檢查當前使用者是否已登入
        val user1 = auth.currentUser
        if (user1 != null) {
            // 取得使用者的 UID
            val uid = user1.uid

// 查詢點擊次數 number
            val client = OkHttpClient()
            val urlBuilder =
                "https://fangbibi.lionfree.net/app/user_number_select.php".toHttpUrlOrNull()
                ?.newBuilder()
                ?.addQueryParameter("uid", "${uid}")
            val url = urlBuilder?.build().toString()
            val request = Request.Builder()
                .url(url)
                .build()
            client.newCall(request).enqueue(object : Callback {
                override fun onFailure(call: Call, e: IOException)
                {
                    e.printStackTrace()
                }
                override fun onResponse(call: Call, response: Response)
                {
                    if (response.isSuccessful)
                    {
                        val responseBody = response.body?.string()
                        val jsonObject = JSONObject(responseBody)
                        val dataArray = jsonObject.getJSONArray("data")
                        runOnUiThread {
                            for (i in 0 until dataArray.length())
                            {
                                val chatObject = dataArray.getJSONObject(i)
                                val number = chatObject.getInt("number")
                                var image: String = "baby"
                                if(number > 9)

```

Homepage::class.java)

```
        {
            image = "lemon3"
        }
        else if(number > 6)
        {
            image = "water2"
        }
        else if(number > 3)
        {
            image = "pandas1"
        }
        else
        {
            image = "baby"
        }
        val bundle = Bundle()
        println(uid)
        bundle.putString("key", uid)
        //bundle.putInt("id", number)
        bundle.putString("image", image)
        val intenthomepage = Intent(this@MainActivity,

intenthomepage.putExtras(bundle)
println(bundle)
println(intenthomepage)
startActivity(intenthomepage)
    }
    }
    }
    }
    })
}
else {
    // 使用者尚未登入，處理未登入的情況
    println("User not logged in")
}
} else {
    it.exception?.message?.let { }
    println("-----error-----")
    showMessage("登入失敗，帳號或 密碼錯誤")
}
```

```

        updateUI(null)
    }
}

//Step 6: 確認更新登入的 user 狀況
private fun updateUI(user: FirebaseUser?) {
    if ( user!= null){
        //已登入
        binding.email.visibility = View.GONE
        binding.pwd.visibility = View.GONE
        binding.signIn.visibility = View.GONE
        binding.signUp.visibility = View.GONE
//        binding.logout.visibility = View.VISIBLE
    }else{
        //未登入
        binding.email.visibility = View.VISIBLE
        binding.pwd.visibility = View.VISIBLE
        binding.signIn.visibility = View.VISIBLE
        binding.signUp.visibility = View.VISIBLE
//        binding.logout.visibility = View.GONE
    }
}

//Step 3-3: 設計登入的 button->設計一個給使用者確認的 message
private fun showMessage(message: String) {
    val alertDialog: AlertDialog.Builder = AlertDialog.Builder(this@MainActivity)
    alertDialog.setMessage(message)
    alertDialog.setPositiveButton("確定") { dialog, which -> }
    alertDialog.show()
}

private fun handleFacebookAccessToken(token: AccessToken) {
    val credential = FacebookAuthProvider.getCredential(token.token)
    auth.signInWithCredential(credential)
        .addOnCompleteListener(this) { task ->
            if (task.isSuccessful) {
// 登入成功
                val user = auth.currentUser
// 可以在這邊處理 facebook 登入後的流程邏輯
            } else {
// 處理登入失敗的狀況，使用 toast 顯示登入失敗的 提示
                Toast.makeText(this@MainActivity, "登入失敗", Toast.LENGTH_SHORT).show()
            }
        }
}

```

```

    }
}
}
}

```

(三)、首頁

介面藍圖(首頁)(activity_homepage.xml)包含使用者資料、使用者發票欄位，介面藍圖(首頁)如圖 8 所示。



圖 8 介面藍圖(首頁)(activity_homepage.xml)

```

package com.example.isopod_invoice

import android.content.Intent
import android.os.Bundle
import android.view.MotionEvent
import android.view.View
import android.view.inputmethod.InputMethodManager
import android.widget.*
import androidx.appcompat.app.AppCompatActivity
import com.google.android.material.bottomnavigation.BottomNavigationView
import com.google.android.material.snackbar.Snackbar
import okhttp3.*

```

```

import okhttp3.HttpUrl.Companion.toHttpUrlOrNull
import org.json.JSONObject
import java.io.IOException

class Homepage : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_homepage)

        // 頁面轉換 Start
        val bottomNavigationView=findViewById<BottomNavigationView>(R.id.bottomNavigationView)
        bottomNavigationView.selectedItemId = R.id.Homepage
        bottomNavigationView.setOnNavigationItemSelectedListener(BottomNavigationView.OnNavigationItemSelectedListener { item ->
            when (item.itemId) {
                R.id.Mine -> {
                    intent?.extras?.let {
                        val uid = it.getString("key")
                        val bundle = Bundle()
                        println(uid)
                        bundle.putString("key", uid)
                        var intentmine = Intent(this,Mine::class.java).apply {
                            putExtras(bundle)
                        }
                        println(bundle)
                        println(intentmine)
                        startActivity(intentmine)
                        overridePendingTransition(0, 0)
                        return@OnNavigationItemSelectedListener true
                    }
                }
                R.id.Homepage -> return@OnNavigationItemSelectedListener true
                R.id.Info -> {
                    intent?.extras?.let {
                        val uid = it.getString("key")
                        val bundle = Bundle()
                        println(uid)
                        bundle.putString("key", uid)
                        var intentinfo = Intent(this, Info::class.java).apply {
                            putExtras(bundle)
                        }
                    }
                }
            }
        })
    }
}

```

```

        println(bundle)
        println(intentinfo)
        startActivity(intentinfo)
        overridePendingTransition(0, 0)
        return@OnNavigationItemSelectedListener true
    }
}
false
})
//頁面轉換 End
//<-----點桌面鍵盤消失-----S>
// 獲取最外層的 View
val rootView: View = findViewById(android.R.id.content)
// 設置畫面點擊事件監聽器
rootView.setOnTouchListener { v, event ->
    // 判斷點擊事件的類型
    if (event.action == MotionEvent.ACTION_DOWN) {
        // 獲取輸入法管理器
        val inputMethodManager =
            getSystemService(INPUT_METHOD_SERVICE) as InputMethodManager
        // 隱藏鍵盤
        inputMethodManager.hideSoftInputFromWindow(v.windowToken, 0)
    }
    false
}
//<-----點桌面鍵盤消失-----E>
val yearmonth = findViewById<EditText>(R.id.YearMonth)
val number = findViewById<EditText>(R.id.InvoiceNumber)
val Check = findViewById<Button>(R.id.CheckButton)
val baby = findViewById<pl.droidsonroids.gif.GifImageView>(R.id.baby)
val pandas1 = findViewById<pl.droidsonroids.gif.GifImageView>(R.id.pandas1)
val water2 = findViewById<pl.droidsonroids.gif.GifImageView>(R.id.water2)
val lemon3 = findViewById<pl.droidsonroids.gif.GifImageView>(R.id.lemon3)
//
    var sopodnumber = 0
//確定 user 的鼠婦為哪一階級
//可以直接變成用 Sql 查詢，不要使用 bundle 傳送
intent?.extras?.let {
    val uid = it.getString("key")
    val clientid = OkHttpClient()

```

```

        val urlBuilderid =
"https://fangbibi.lionfree.net/app/user_number_select.php".toHttpUrlOrNull()
        ?.newBuilder()
        ?.addQueryParameter("uid", "${uid}")
    val urlid = urlBuilderid?.build().toString()
    val requestid = Request.Builder()
        .url(urlid)
        .build()
    clientid.newCall(requestid).enqueue(object : Callback {
        override fun onFailure(call: Call, e: IOException)
        {
            e.printStackTrace()
        }
        override fun onResponse(call: Call, response: Response)
        {
            if (response.isSuccessful)
            {
                val responseBodyid = response.body?.string()
                //println("id-----${responseBody}")
                val jsonObjectid = JSONObject(responseBodyid)
                val dataArrayid = jsonObjectid.getJSONArray("data")
                runOnUiThread {
                    for (i in 0 until dataArrayid.length())
                    {
                        val chatObjectid = dataArrayid.getJSONObject(i)
                        val sopodnumberid = chatObjectid.getInt("number")
                        //println(sopodnumber)
                        if(sopodnumberid > 9)
                        {
                            baby.visibility = View.INVISIBLE
                            pandas1.visibility = View.INVISIBLE
                            water2.visibility = View.INVISIBLE
                            lemon3.visibility = View.VISIBLE
                        }
                        else if(sopodnumberid > 6)
                        {
                            baby.visibility = View.INVISIBLE
                            pandas1.visibility = View.INVISIBLE
                            water2.visibility = View.VISIBLE
                            lemon3.visibility = View.INVISIBLE
                        }
                    }
                }
            }
        }
    })

```

```

    }
    else if(sopodnumberid > 3)
    {
        baby.visibility = View.INVISIBLE
        pandas1.visibility = View.VISIBLE
        water2.visibility = View.INVISIBLE
        lemon3.visibility = View.INVISIBLE
    }
    else
    {
        baby.visibility = View.VISIBLE
        pandas1.visibility = View.INVISIBLE
        water2.visibility = View.INVISIBLE
        lemon3.visibility = View.INVISIBLE
    }
}
}
}
}
}}
})
//確定 user 鼠婦 End
//Check Button Start
Check.setOnClickListener {
    if((yearmonth.text.isNotEmpty()) and (number.text.isNotEmpty()))
    {
        val vid = number.text.toString()
        println("cid-----${vid}")
        val postyearmonth = yearmonth.text.toString()
        val parts = postyearmonth.split("-");//分割 yearmonth
        val year = parts[0]
        val month = parts[1]
        if(number.length() == 8)
        {
            //-----
            intent?.extras?.let {
                val uid = it.getString("key")
                val clientPost = OkHttpClient()
                val requestBody = FormBody.Builder()
                    .add("uid", "${uid}")
                    .add("vid", "${vid}")

```



```

        .add("vmon", "${month}")
        .add("vyear", "${year}")
        .build()
val requestPost = Request.Builder()
    .url("https://fangbibi.lionfree.net/app/invoiceinsert.php")
    .post(requestBody)
    .build()
clientPost.newCall(requestPost).enqueue(object : Callback
{
    override fun onFailure(call: Call, e: IOException)
    {
        e.printStackTrace()
    }
    override fun onResponse(call: Call, response: Response)
    {
        val responseBody = response.body?.string()
        // Handle the response body
        println("POST-----${responseBody}")
        runOnUiThread { }

//Snackbar
        val snackbar = Snackbar.make(Check, "${responseBody}" ,
Snackbar.LENGTH_SHORT)

        snackbar.show()

// 鼠婦長大 End
    }
})

//更新鼠婦
val client = OkHttpClient()
val urlBuilder =
"https://fangbibi.lionfree.net/app/user_number_select.php".toHttpUrlOrNull()
    ?.newBuilder()
    ?.addQueryParameter("uid", "${uid}")
val url = urlBuilder?.build().toString()
val request = Request.Builder()
    .url(url)
    .build()
client.newCall(request).enqueue(object : Callback
{
    override fun onFailure(call: Call, e: IOException)
    {

```

```

        e.printStackTrace()
    }
    override fun onResponse(call: Call, response: Response)
    {
        if (response.isSuccessful)
        {
            val responseBody = response.body?.string()
            val jsonObject = JSONObject(responseBody)
            val dataArray = jsonObject.getJSONArray("data")
            runOnUiThread {
                for (i in 0 until dataArray.length())
                {
                    val chatObject = dataArray.getJSONObject(i)
                    val sopodnumber = chatObject.getInt("number")
                    if(sopodnumber > 9)
                    {
                        baby.visibility = View.INVISIBLE
                        pandas1.visibility = View.INVISIBLE
                        water2.visibility = View.INVISIBLE
                        lemon3.visibility = View.VISIBLE
                    }
                    else if(sopodnumber > 6)
                    {
                        baby.visibility = View.INVISIBLE
                        pandas1.visibility = View.INVISIBLE
                        water2.visibility = View.VISIBLE
                        lemon3.visibility = View.INVISIBLE
                    }
                    else if(sopodnumber > 3)
                    {
                        baby.visibility = View.INVISIBLE
                        pandas1.visibility = View.VISIBLE
                        water2.visibility = View.INVISIBLE
                        lemon3.visibility = View.INVISIBLE
                    }
                    else
                    {
                        baby.visibility = View.VISIBLE
                        pandas1.visibility = View.INVISIBLE
                        water2.visibility = View.INVISIBLE

```

```

lemon3.visibility = View.INVISIBLE
    }
    }
    }
    }
    })
//更新 End
    }
}
else
{
    Toast.makeText(this,"請確認發票號碼是否為 8 碼", Toast.LENGTH_SHORT).show()
}
}
else{
    Toast.makeText(this,"請輸入資料", Toast.LENGTH_SHORT).show()
}
}
//Check Button End
}
}

```

(四)、使用者發票資訊

介面藍圖(使用者發票資訊)(activity_mine.xml)包含使用者發票、開獎發票欄位，介面藍圖(使用者發票資訊)如圖所示。



圖 9 介面藍圖(使用者發票資訊)(activity_mine.xml)

```
package com.example.isopod_invoice

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.text.method.ScrollingMovementMethod
import android.widget.Button
import android.widget.TextView
import android.widget.Toast
import com.google.android.material.bottomnavigation.BottomNavigationView
import okhttp3.*
import okhttp3.HttpUrl.Companion.toHttpUrlOrNull
import org.json.JSONObject
import java.io.IOException

class Mine : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
```

```

        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_mine)
//頁面轉換 Start
        val bottomNavigationView=findViewById<BottomNavigationView>(R.id.bottomNavigationView)
        bottomNavigationView.selectedItemId = R.id.Homepage
bottomNavigationView.setOnItemClickListener(BottomNavigationView.OnNavigationItemSelectedListener
SelectedListener { item ->
    when (item.itemId) {
        R.id.Mine -> return@OnNavigationItemSelectedListener true
        R.id.Homepage -> {
            intent?.extras?.let {
                val uid = it.getString("key")
                val bundle = Bundle()
                println(uid)
                bundle.putString("key", uid)
                var intenthomepage = Intent(this,Homepage::class.java).apply {
                    putExtras(bundle)
                }
                println(bundle)
                println(intenthomepage)
                startActivity(intenthomepage)
                overridePendingTransition(0, 0)
                return@OnNavigationItemSelectedListener true
            }
        }
        R.id.Info -> {
            intent?.extras?.let {
                val uid = it.getString("key")
                val bundle = Bundle()
                println(uid)
                bundle.putString("key", uid)
                var intentinfo = Intent(this,Info::class.java).apply {
                    putExtras(bundle)
                }
                println(bundle)
                println(intentinfo)
                startActivity(intentinfo)
                overridePendingTransition(0, 0)
                return@OnNavigationItemSelectedListener true
            }
        }
    }
}

```

```

        false
    })
//頁面轉換 End
//Show 自己的所有發票 Start
val Show = findViewById<Button>(R.id.Show)
Show.setOnClickListener {
    intent?.extras?.let {
        val uid = it.getString("key")
        val client = OkHttpClient()
        val urlBuilder =
"https://fangbibi.lionfree.net/app/user_invoice_check.php".toHttpUrlOrNull()
        ?.newBuilder()
        ?.addQueryParameter("uid", "${uid}")
        val url = urlBuilder?.build().toString()
        val request = Request.Builder()
            .url(url)
            .build()
        println(request.url.toString())
        client.newCall(request).enqueue(object : Callback {
            override fun onFailure(call: Call, e: IOException) {
                e.printStackTrace()
            }
            override fun onResponse(call: Call, response: Response) {
                if (response.isSuccessful) {
                    val responseBody = response.body?.string()
                    val jsonObject = JSONObject(responseBody)
                    val dataArray = jsonObject.getJSONArray("data")
                    runOnUiThread {
                        //直接將內容回傳給 id 名稱為 TextView
                        val textView = findViewById<TextView>(R.id.Information)
textView.setMovementMethod(ScrollingMovementMethod.getInstance())
                        if (dataArray.length() == 0) {
                            textView.text = "查無資料"
                        } else {
                            textView.text = " 發票號碼      年      月      狀態
金額\n"

                            for (i in 0 until dataArray.length()) {
                                val chatObject = dataArray.getJSONObject(i)
                                val vid = chatObject.getString("vid")
                                val vmon = chatObject.getString("vmon")

```

```

        val vyear = chatObject.getInt("vyear")
        val statue = chatObject.getString("statue")
        val price = chatObject.getString("price")
        val formattedText = String.format(
            "%-11s %-8d %-6s %-10s %8s\n",
            vid, vyear, vmon, statue, price
        )
        textView.append(formattedText)
    }
}
} else {
    println("Request failed")
    runOnUiThread {
        //直接將內容回傳給 id 名稱為 TextView
        findViewById<TextView>(R.id.Information).text = "資料錯誤"
    }
}
})}
}
//Show 自己的所有發票 End
}
}

```

(五)、開獎發票資訊

介面藍圖(開獎發票資訊)(activity_info.xml)包含開獎發票欄位，介面藍圖(開獎發票資訊)如圖 10 所示。



圖 10 介面藍圖(開獎發票資訊)(activity_info.xml)

```
package com.example.isopod_invoice

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.text.method.ScrollingMovementMethod
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import android.widget.Toast
import com.google.android.material.bottomnavigation.BottomNavigationView
import okhttp3.*
import okhttp3.HttpUrl.Companion.toHttpUrlOrNull
import org.json.JSONObject
import java.io.IOException

class Info : AppCompatActivity() {
```



```

        override fun onCreate(savedInstanceState: Bundle?) {
            super.onCreate(savedInstanceState)
            setContentView(R.layout.activity_info)
//頁面轉換 Start
            val bottomNavigationView=findViewById<BottomNavigationView>(R.id.bottomNavigationView)
            bottomNavigationView.selectedItemId = R.id.Homepage
            bottomNavigationView.setOnNavigationItemSelectedListener(BottomNavigationView.OnNavigationItemSelectedListener { item ->
                when (item.itemId) {
                    R.id.Mine -> {
                        intent?.extras?.let {
                            val uid = it.getString("key")
                            val bundle = Bundle()
                            println(uid)
                            bundle.putString("key", uid)
                            var intentmine = Intent(this, Mine::class.java).apply {
                                putExtras(bundle)}
                            println(bundle)
                            println(intentmine)
                            startActivity(intentmine)
                            overridePendingTransition(0, 0)
                            return@OnNavigationItemSelectedListener true
                        }
                    }
                    R.id.Homepage -> {
                        intent?.extras?.let {
                            val uid = it.getString("key")
                            val bundle = Bundle()
                            println(uid)
                            bundle.putString("key", uid)
                            var intenthomepage = Intent(this, Homepage::class.java).apply {
                                putExtras(bundle)}
                            println(bundle)
                            println(intenthomepage)
                            startActivity(intenthomepage)
                            overridePendingTransition(0, 0)
                            return@OnNavigationItemSelectedListener true
                        }
                    }
                    R.id.Info -> return@OnNavigationItemSelectedListener true
                }
            })
            false

```

```

    })
//頁面轉換 End

    val Show = findViewById<Button>(R.id.ShowButton)
    val yearmonth = findViewById<EditText>(R.id.yearmonth)
    Show.setOnClickListener {
        if (yearmonth.text.isNotEmpty()){
            val postyearmonth = yearmonth.text.toString()
            val parts = postyearmonth.split("-");//分割 yearmonth
            val year = parts[0]
            val month = parts[1]
            val client = OkHttpClient()
            val urlBuilder =
"https://fangbibi.lionfree.net/app/testread.php".toHttpUrlOrNull()
                ?.newBuilder()
                ?.addQueryParameter("month", "${month}")
                ?.addQueryParameter("year", "${year}")
            val url = urlBuilder?.build().toString()
            val request = Request.Builder()
                .url(url)
                .build()
            println(request.url.toString())
            client.newCall(request).enqueue(object : Callback {
                override fun onFailure(call: Call, e: IOException) {
                    e.printStackTrace()
                }
                override fun onResponse(call: Call, response: Response) {
                    if (response.isSuccessful) {
                        val responseBody = response.body?.string()
                        val jsonObject = JSONObject(responseBody)
                        val dataArray = jsonObject.getJSONArray("data")
                        runOnUiThread {
                            //直接將內容回傳給 id 名稱為 TextView
                            val textView = findViewById<TextView>(R.id.Information)
                            if (dataArray.length() == 0) {
                                textView.text = "查無資料，發票還未開獎或已過期 6 月以上"
                            } else {
                                var month_1 = month.toInt()+1;
                                textView.setText(month_1.toString())
                            }
                        }
                    }
                }
            })
            textView.setMovementMethod(ScrollingMovementMethod.getInstance());
        }
    }
}

```

```

        textView.text = "${year} - ${month}、${month_1}月 \n 開獎號碼    獎項 \n"

        for (i in 0 until dataArray.length()) {
            val chatObject = dataArray.getJSONObject(i)
            val cid = chatObject.getString("cid")
            val cmon = chatObject.getInt("cmon")
            val cyear = chatObject.getInt("cyear")
            val cprice = chatObject.getString("cprice")
            val cdate = chatObject.getString("cdate")
            textView.append("${cid} - ${cprice}\n")
        }
    }
} else {
    println("Request failed")
    runOnUiThread {
        findViewById<TextView>(R.id.Information).text = "資料錯誤"
    }
}

}))
}
else{
    Toast.makeText(this,"請輸入資料", Toast.LENGTH_SHORT).show()
}
}
}

```

註冊(A)	登入(B)	首頁(C)	使用者發票資訊(D)	開獎發票資訊(E)
註冊(A1)	登入(B1)	新增發票(C1)	查看使用者發票(D1)	開獎發票資訊(E1)
		角色飼養機制(C2)	發票兌獎(D2)	

圖 11 階層式介面架構(系統功能表)

三、開發工具

- 程式語言：Kotlin, PHP, JSON, XML
- IDE：Android Studio, firebase, lionfree
- 資料庫系統：MySQL