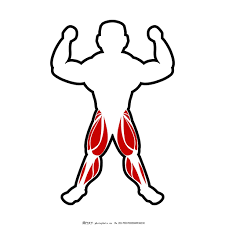
**國立台北商業大學**

**資訊管理系**

**110年度專案設計**

**系統手冊**



**健**

**康**

組別：N110402

題目：年長者肌力測試

指導老師：唐日新 老師

組長：N1076411康博閎

組員：N1076405張勝崴 N1076414夏聖捷

N1076406江昱霆 N1076448郭金樑

中華民國110年 11月 30日

**目錄**

1. **背景與動機**
   1. 簡介
   2. 問題與機會
   3. 相關系統探討
2. **系統目標與預期成果**

2-1 系統目標

2-2 預期成果

1. **系統規格**

3-1 系統架構：最好以圖示方式說明

3-2 系統軟、硬體需求與技術平台

3-3 開發標準與使用工具

1. **專案時程與分工**

4-1 專案時程

4-2 專案分工

1. **需求模型**

5-1 使用者需求

5-2 使用個案圖(Use case diagram)

5-3使用活動圖(Activity diagram)

5-4分析類別圖(Analysis class diagram)

分析物件圖(Analysis object diagram)

1. **設計模型**

6-1循序圖(Sequential diagram)

6-2設計類別圖(Design class diagram)

設計物件圖(Design object diagram)

1. **實作模型**

7-1佈署圖(Deployment diagram)。

7-2套件圖(Package diagram)。

7-3元件圖(Component diagram)。

7-4狀態機(State machine)

甚至時序圖(Timing diagram)

1. **資料庫設計**
2. **程式**

9-1元件清單及其規格描述

9-2其他附屬之各種元件

1. **測試模型**

10-1測試計畫

10-2測試個案與測試結果資料

**第十一章、操作手冊**

**第十二章、使用手冊**

**第十三章、感想**

**第十四章、參考資料**

**附錄 審查評審意見之修正情形**

1. 背景與動機

1-1簡介

　　隨著衛生醫療、生化科技快速進步，國人平均壽命也逐漸延長，進而影響人口結構與經濟層面。依據聯合國世界衛生組織 (World Trade Organization，簡稱 WTO)的定義，當一個地區中高齡人口超過總人口數的7%時，即為高齡化社會。臺灣地區高齡人口在1993年9月已經超過總人口數的7%，因此早己經是高齡化社會。

老化為人體正常的現象；但隨著年齡增長，人體各項機能也會逐漸退化，進而產生許多問題，而跌倒便是其中之一。而當老人或高齡者產生跌倒後往往會產生更多耗費國家與社會醫療資源的問題，如髖關節骨折、其他部位骨折、硬腦膜下出血、其他嚴重的軟組織傷害、頭部外傷甚至是造成死亡等。若要歸究原因，下肢運動能力的衰退和高齡者容易產生跌倒是有其關連性的。

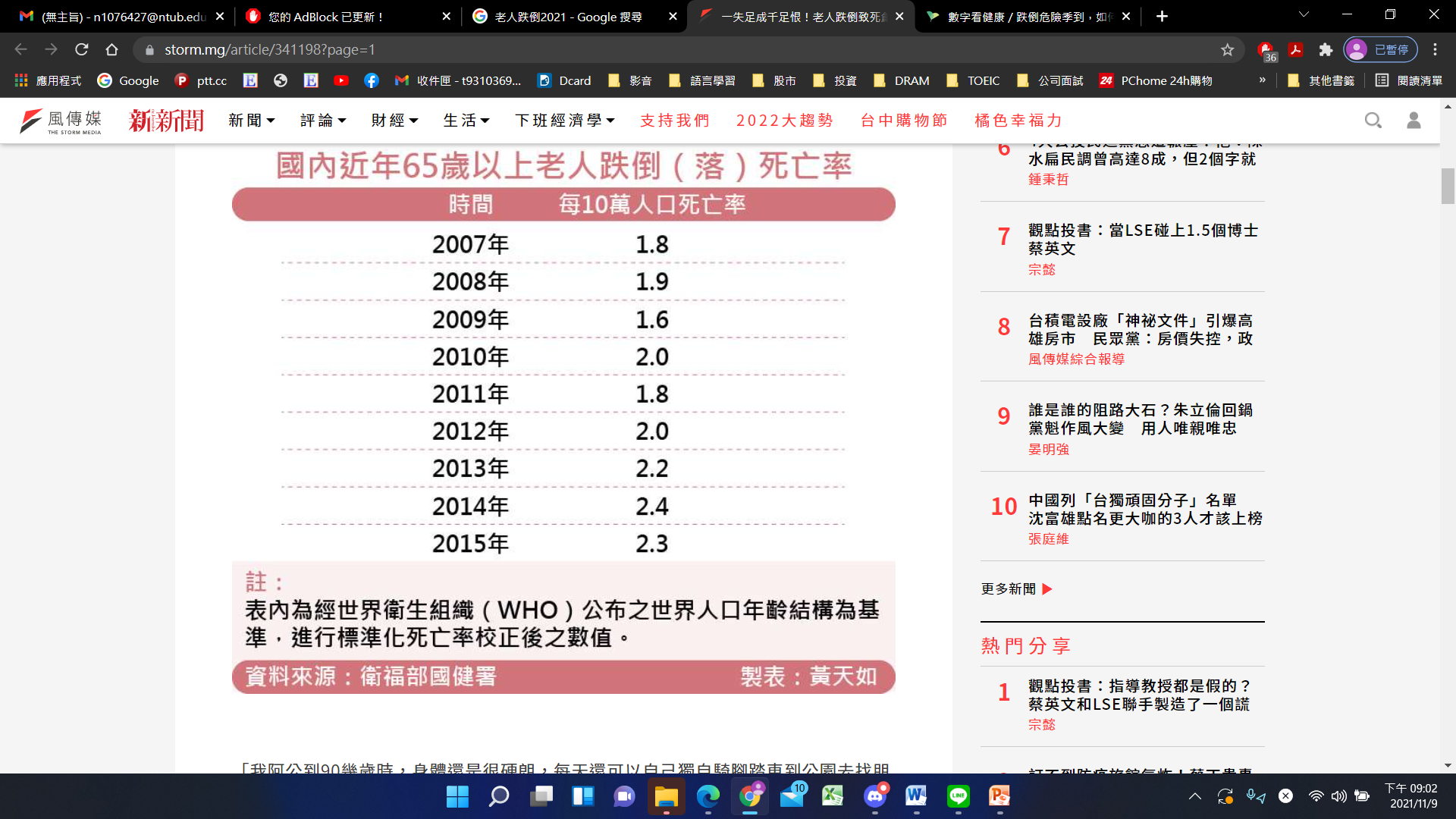
隨著社會與經濟的發展，生活品質也逐漸為國人所注重，健康的生活更是高齡者最為關心的話題。就人體而言，肌肉的收縮可以讓身體各肢段產生運動，而這樣的運動則可以讓我們能夠從事各項的身體活動。在運動的領域中，運動選手需要肌力訓練來增加自己的肌肉力量進而有優秀之成績。就一般人而言，無論是成年男女或者是青少年則可接受適當的重量訓練來強健體魄、增進體能、改善個人外觀還有促進身體力量的發展等。

下肢肌力訓練對高齡者下肢運動表現及健康生活形態之影響。根據美國運動醫學學會所提出的觀點，肌力訓練可以協助加強高齡者的活動能力，防止因為老化所引起的肌肉流失、低骨質密度和低心血管循環能力，另外也可強化高齡者的平衡能力和步態等。

高齡化所導致的下肢運動能力衰退則容易造成老年人產生跌倒的傷害並且使國家社會耗費更多的醫療資源。

現在市面上有販售檢測儀器，但儀器價格昂貴，而且有些儀器操作複雜沒有他人輔助測量，對老人家是非常不友善的，且其測量結果是否精準，也是依實際情況來決定，醫院雖然有儀器可以精準測量，但考量地理位置、交通、預約掛號繳費等繁雜手續，所以有很多家庭種種因素而直接放棄。

　　牆滑梯(Wall Slides)的運動動作，能提高股四頭肌、肌大臂和小腿的力量，這個動作主要是拿來訓練下肢肌力，也被物理自療師用來作為評估工具，只作為較為粗略的測驗，並不能較為精確的得操作者彎曲的角度。參考上述，所以我們開發此APP，免費且操作方便只需要花少許時間就可以檢測完成並儲存測試數據，可以快速了解到家中長輩的腿部肌耐力的狀況，假如測試狀況不理想至少有個概念，可以提早防範，避免遺憾發生。



圖一

1-2問題與機會

**SWOT分析**

|  |  |
| --- | --- |
| **S優勢** | **W劣勢** |
| ˙ APP免費下載  ˙ 操作方便  ˙ 簡易測量 | ˙ 只針對特定部位  ˙ 知名度不夠  ˙ 數據公信力不足 |
| **O機會** | **T威脅** |
| ˙ 市面上測試軟體極為少數  ˙˙社會高齡化，需求人群增加  ˙‧部分取代醫療專業儀器功能 | ˙ 沒有廣告，推廣不易  ˙˙無專業醫療儀器準確 |

1-3相關系統探討

|  |  |  |
| --- | --- | --- |
| APP  功能 | 年長者肌力測試 | 全身鍛鍊-在家運動 |
| 定位 | 年長者為主 | 大眾 |
| 功能 | 腿部肌力測試 | 肌群的訓練 |
| 目的 | 了解腿部肌力是否足夠 | 鍛鍊出完美身材 |
| 評價 | 無 | 由使用者評價 |
| 反饋 | 無 | 由使用者反饋 |

第二章、系統目標與預期成果

2-1系統目標

　　此APP主要針對高齡族群。隨著成年人身體隨著年齡的逐漸增長，腿部肌力開始慢慢下降，引發了許多意外事故，開發此APP主要讓大家重視老人的身體，用這個系統檢測老年的腿部肌耐力，讓他們了解到老年人是否需要加強腿部的運動，防止更多意外發生。

1. 降低測量肌力成本

2. 測量資訊儲存做初步分析或供醫生參考

3. 提高測量方便性，不用跑醫院或復健中心

4. 針對大腿肌少症提出建議

**APP功能：**

1. 腿部的肌力測試
2. 測試數據儲存
3. 提供簡單數據分析是否有就醫需求

2-2預期成果

　　透過此APP可以幫助需要肌耐力測試的老年人，讓他們了解到自己腿部肌力處於何種程度，是否需要進行加強，避免意外發生。

第三章、系統規格

3-1 系統架構

3-2 系統軟、硬體需求與技術平台

|  |  |
| --- | --- |
| 軟體需求 | |
| 手機作業系統 | Android 4.0 以上 |
| 硬體需求 | |
| 手機硬體需求 | CPU 1GHz 以上 |
| 記憶體 512M 以上 |
| 剩餘空間 16MB |

3-3 使用標準與工具

|  |  |  |
| --- | --- | --- |
| 類型 | 工具名稱 | 符號 |
| 程式語言 | android studio |  |
| Java |  |
| 資料庫伺服器 | SQLite |  |
| 相關軟體 | MPchart | 折線圖 |
| Diagrams.net |  |
| Microsoft office |  |

第四章、專案時程與分工

4-1 專案時程

預定進度

實際進度

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 年度 | 110 | | | | | | | | | | |
| 項目/月份 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 主題制定 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| 資料蒐集 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| 系統規劃 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| 系統分析 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| 程式設計 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| 內文撰寫 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| Logo 繪製 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| 系統測試 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| 口頭簡報 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

4-2 專業分工

●：主要負責

○：協助負責

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 組員  工作項目 | | 張勝崴 | 江昱霆 | 康博閎 | 夏聖捷 | 郭金樑 |
| 系統開發 前作業 | 主題構思 | ● | ● | ● | ● | ● |
| 資料蒐集 | ● | ● | ● | ● | ● |
| 系統架構、功能 | ● | ● | ● | ● | ● |
| 美工 | 介面設計 | ● |  |  |  | ● |
| LOGO  設計 |  |  | ● |  |  |
| 程式 | 程式設計 | ● |  |  |  |  |
| 程式測試 | ● | ● | ● | ● | ● |
| 系統維護 | ● |  |  |  |  |
| 文書 | 文案撰寫 |  |  | ○ |  | ● |
| UML 繪製 | ○ |  | ○ |  | ● |
| 統整 |  |  |  |  | ● |
| 專題發表 | 簡報構思 與製作 |  | ● | ● | ● | ● |

1. 需求模型

5-1 使用者需求

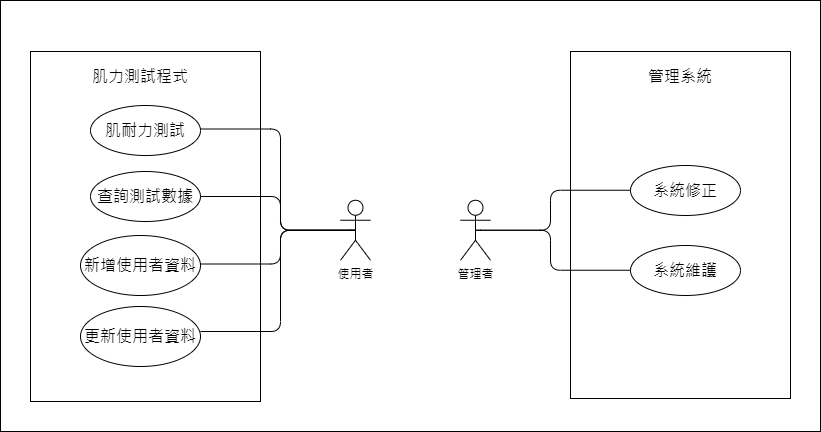
表 5-1-1 功能需求表

|  |  |
| --- | --- |
| 功能需求 | |
| 需求 | Use Case |
| 使用者想要進行測試 | 肌力測試系統 |
| 使用者需要查看紀錄 | 歷史紀錄 |
| 使用者想要新增資料 | 使用者資料管理 |
| 使用者想要更改資料 | 使用者資料管理 |

表 5-1-2 非功能需求

|  |  |
| --- | --- |
| 非功能需求 | |
| 需求 | Use Case |
| 需使用Android系統 | — |

5-2 使用個案圖Use Case

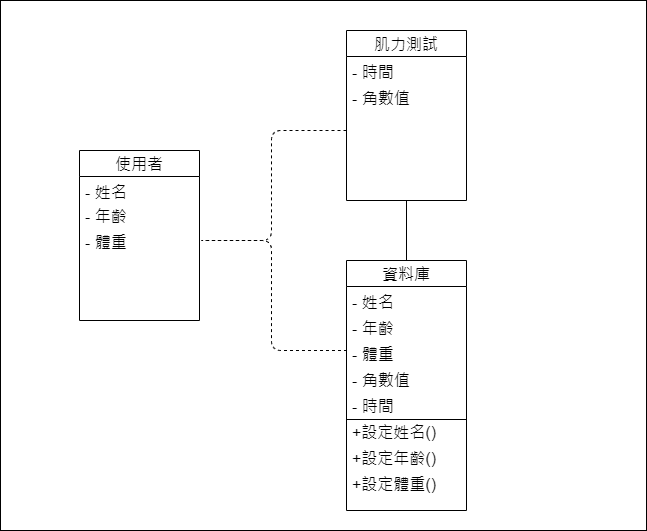


5-3 使用個案描述

|  |
| --- |
| 圖5-3-1系統活動圖 |
|  |
| 圖5-3-2肌力測試活動圖 |
|  |

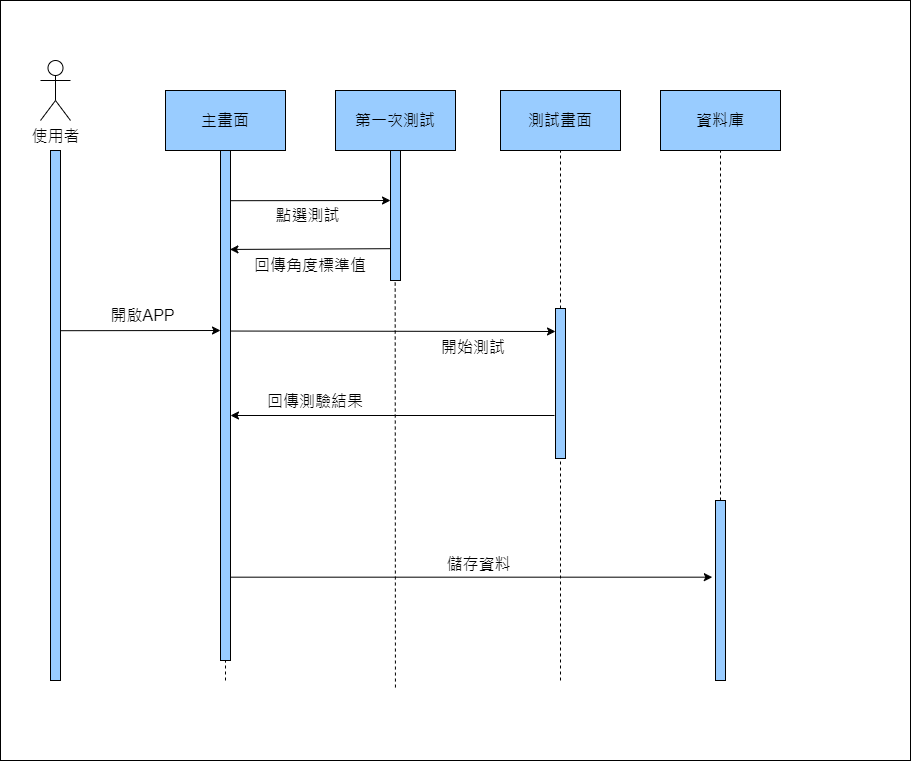
|  |
| --- |
| 圖5-3-3使用者資料活動圖 |
|  |

* 1. 分析類別圖

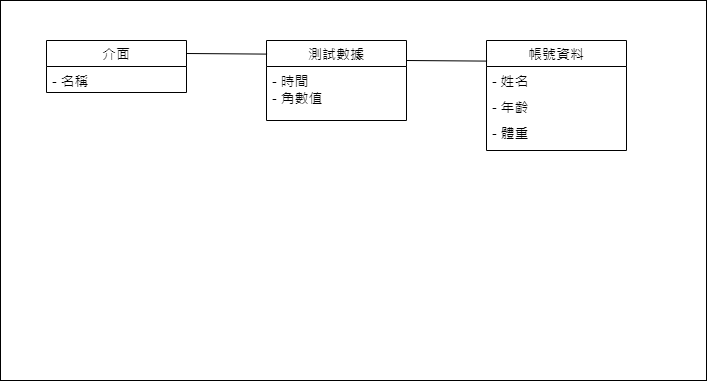


第六章、設計模型

6-1 循序圖

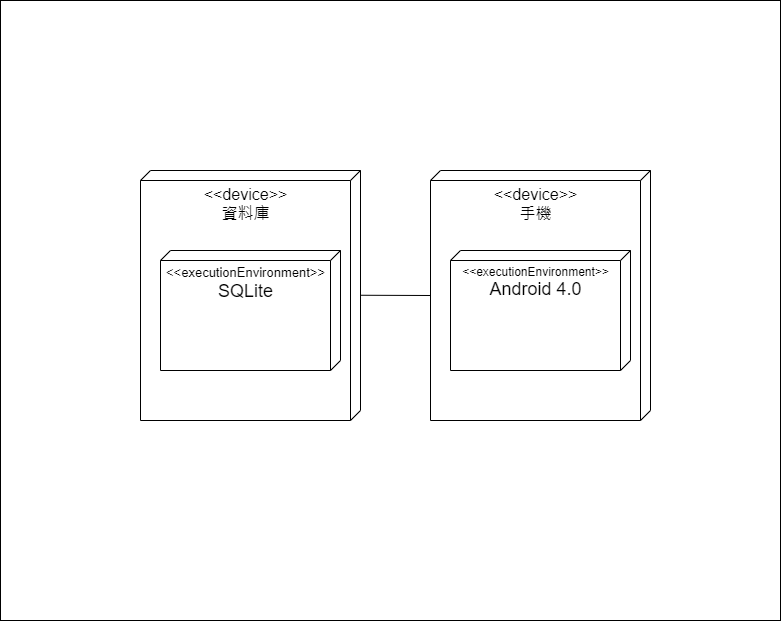


* 1. 設計類別圖

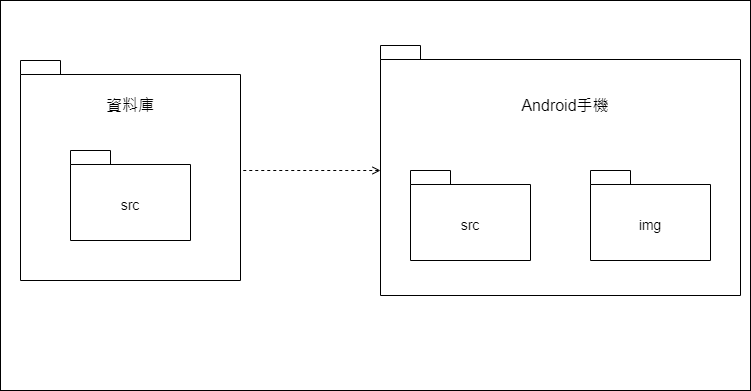


第七章、實作模型

7-1 部署圖



7-2 套件圖



* 1. 狀態圖

|  |
| --- |
| 圖7-3-1第一次測試 |
|  |

|  |
| --- |
| 圖7-3-2測試畫面 |
|  |

第八章、資料庫設計

表 8-1 資料表清單

|  |  |
| --- | --- |
| 編號 | 資料表名稱 |
| A01 | DBhelper |

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | A01 | 資料表名稱 | DBhelper |
| 資料表功能 | 儲存時間、成功次數、姓名、體重、年齡 | | |
| 欄位名稱 | 資料型態 | 欄位說明 | 允許為NULL值 |
| name | String | 儲存使用者姓名 | No |
| wid | Float | 儲存體重 | No |
| old | Int | 儲存年齡 | No |
| stime | Int | 儲存成功次數 | No |
| time | String | 儲存測試花費時間 | No |

第九章、程式

9-1 元件清單及其規格描述

表 9-1 元件清單及其規格描述 編號 程式名稱 元件模組

|  |  |  |
| --- | --- | --- |
| 編號 | 程式設計 | 元件模組功能 |
| 1 | Main\_activity | 主頁面 |
| 2 | Text | 測試頁面 |
| 3 | Record | 查詢頁面 |
| 4 | ChartLinerManager | 折線圖形化 |
| 5 | DBHelper | SQLite |

9-2 其他附屬之各種元件

表 9-2-1 其他元件清單

|  |  |
| --- | --- |
| MPChart | |
| 用途 | 折線圖形化 |
| 參數 | 無 |
| 傳回值 | 無 |
| 所屬類別 | MPchart.class |

|  |  |
| --- | --- |
| SQL | |
| 用途 | 資料庫 |
| 參數 | 無 |
| 傳回值 | 無 |
| 所屬類別 | SQLite.class |

|  |  |
| --- | --- |
| Sensor | |
| 用途 | 陀螺儀傳感器 |
| 參數 | 傳感器角速度 |
| 傳回值 | 角度值 |
| 所屬類別 | SQLite.class |

表 9-2-2 程式碼

|  |
| --- |
| Main |
| package com.example.axix;  import androidx.appcompat.app.AlertDialog;  import androidx.appcompat.app.AppCompatActivity;  import android.content.Context;  import android.content.Intent;;  import android.database.Cursor;  import android.graphics.Color;  import android.os.Bundle;  import android.os.Vibrator;  import android.view.View;  import android.widget.Button;  import android.widget.EditText;  import android.widget.TextView;  import android.widget.Toast;  import com.github.mikephil.charting.charts.LineChart;  import java.util.ArrayList;  import java.util.List;  public class MainActivity extends AppCompatActivity {  public Button btnfirsttry, btntest, btnview, howto, view, view1;  public TextView bigax, tbigax;  public EditText timesset;  SaveAngle SA;  SaveTime ST;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);  Vibrator v = (Vibrator) getSystemService(Context.VIBRATOR\_SERVICE);  timesset = (EditText) findViewById(R.id.timesset);  bigax = (TextView) findViewById(R.id.bigax);  tbigax = (TextView) findViewById(R.id.tbigax);  SA = new SaveAngle(this);  ST = new SaveTime(this);  //測試完成接收值  Bundle bundle = getIntent().getExtras();  if (bundle != null) {  float bigestax = bundle.getFloat("bigestax");  int times = bundle.getInt("times");  int rcount = bundle.getInt("fcount");  bigax.setText(String.valueOf(bigestax));  timesset.setText(String.valueOf(times));  tbigax.setText(String.valueOf(rcount));  }  String w = bigax.getText().toString();  float y = Float.parseFloat(w);  String x = tbigax.getText().toString();  int z = Integer.parseInt(x);  //進入第一次嘗試頁面  btnfirsttry = (Button) findViewById(R.id.btnfirsttry);  btnfirsttry.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View f) {  Intent first = new Intent();  first.setClass(MainActivity.this, firsttry.class);  startActivity(first);  //v.vibrate(300);  }  });  //進入測試頁面並重送角度條件  btntest = (Button) findViewById(R.id.btntest);  btntest.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View t) {  Intent test = new Intent(MainActivity.this, test.class);  float bigestax = Float.parseFloat(bigax.getText().toString());  test.putExtra("bigestax", bigestax);  int times = Integer.parseInt(timesset.getText().toString());  test.putExtra("times", times);  startActivity(test);  }  });  //進入記錄頁面  btnview = (Button) findViewById(R.id.btnview);  btnview.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View f) {  Intent re = new Intent(MainActivity.this, record.class);  re.putExtra("ads", bigax.getText().toString() + " " + timesset.getText().toString());  startActivity(re);  }  });  //進入使用說明頁面  howto = (Button) findViewById(R.id.howto);  howto.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View f) {  Intent how = new Intent();  how.setClass(MainActivity.this, Howtouse.class);  startActivity(how);  }  });  view = findViewById(R.id.view);  view.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  Cursor Mres = ST.Tgetdata();  if (Mres.getCount()==0){  Toast.makeText(MainActivity.this, "查無資料", Toast.LENGTH\_SHORT).show();  return;  }  StringBuffer Mbuffer = new StringBuffer();  while (Mres.moveToNext()) {  Mbuffer.append("id :"+Mres.getString(0)+"\n");  Mbuffer.append("時間 :"+Mres.getString(1)+"\n\n");  }  AlertDialog.Builder Mbuilder = new AlertDialog.Builder(MainActivity.this);  Mbuilder.setCancelable(true);  Mbuilder.setTitle("時間紀錄");  Mbuilder.setMessage(Mbuffer.toString());  Mbuilder.show();  }  });  view1 = findViewById(R.id.view1);  view1.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  Cursor Ares = SA.Agetdata();  if (Ares.getCount()==0){  Toast.makeText(MainActivity.this, "查無資料", Toast.LENGTH\_SHORT).show();  return;  }  StringBuffer Abuffer = new StringBuffer();  while (Ares.moveToNext()) {  Abuffer.append("id :"+Ares.getString(0)+"\n");  Abuffer.append("角度 :"+Ares.getString(1)+"\n\n");  }  AlertDialog.Builder Abuilder = new AlertDialog.Builder(MainActivity.this);  Abuilder.setCancelable(true);  Abuilder.setTitle("角度紀錄");  Abuilder.setMessage(Abuffer.toString());  Abuilder.show();  }  });  }  } |

|  |
| --- |
| Text |
| package com.example.axix;  import androidx.appcompat.app.AppCompatActivity;  import android.content.Intent;  import android.graphics.Color;  import android.hardware.Sensor;  import android.hardware.SensorEvent;  import android.hardware.SensorEventListener;  import android.hardware.SensorManager;  import android.os.Bundle;  import android.os.Handler;  import android.widget.TextView;  import com.github.mikephil.charting.charts.LineChart;  import java.util.ArrayList;  import java.util.List;  public class test extends AppCompatActivity implements SensorEventListener {  //陀螺儀  private TextView tshow;  private SensorManager mSensorManager;  private Sensor macce;  private Sensor mgryo;  private Sensor mmagn;  private static final float NS2S = 1.0f / 1000000000.0f;  private float angle[] = new float[3];  //時間  private float timestamp;  //從主頁面把最大角度帶過來當作次數條件  private TextView limited;  //時間  private Long tstartTime;  //延遲  private Handler handler = new Handler();  //成功次數  private TextView scount, aa;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_test);  //取得目前時間  tstartTime = System.currentTimeMillis();  //設定定時要執行的方法  handler.removeCallbacks(updateTimer);  //設定Delay的時間  handler.postDelayed(updateTimer, 1000);  //賦予R.id  aa = (TextView) findViewById(R.id.textView2);  scount = (TextView) findViewById(R.id.scount);  limited = (TextView) findViewById(R.id.limited);  scount = (TextView) findViewById(R.id.scount);  tshow = (TextView) findViewById(R.id.tshowTextView);  mSensorManager = (SensorManager) getSystemService(SENSOR\_SERVICE);  macce = mSensorManager.getDefaultSensor(Sensor.TYPE\_ACCELEROMETER);  mgryo = mSensorManager.getDefaultSensor(Sensor.TYPE\_GYROSCOPE);  mmagn = mSensorManager.getDefaultSensor(Sensor.TYPE\_MAGNETIC\_FIELD);  }  //計時器  public Runnable updateTimer = new Runnable() {  public void run() {  final TextView time = (TextView) findViewById(R.id.ttimer);  Long spentTime = System.currentTimeMillis() - tstartTime;  //計算目前已過分鐘數  Long minius = (spentTime / 1000) / 60;  //計算目前已過秒數  Long seconds = (spentTime / 1000) % 60;  Long mm = spentTime % 1000;  //顯示時間  time.setText(minius + ":" + seconds + "." + mm);  handler.postDelayed(this, 1000);  }  };  //延遲設定  protected void onResume() {  super.onResume();  mSensorManager.registerListener(this, macce, SensorManager.SENSOR\_DELAY\_GAME);  mSensorManager.registerListener(this, mgryo, SensorManager.SENSOR\_DELAY\_GAME);  mSensorManager.registerListener(this, mmagn, SensorManager.SENSOR\_DELAY\_GAME);  }  //暫停程式時停止執行  protected void onPause() {  super.onPause();  mSensorManager.unregisterListener(this);  }  //讀取SensorEvent資訊  @Override  public void onSensorChanged(SensorEvent sensorEvent) {  //測驗時間  Long spentTime = System.currentTimeMillis() - tstartTime;  //從主頁面接收角度條件並轉成值  Bundle bundle = getIntent().getExtras();  if (bundle != null) {  float bigestax = bundle.getFloat("bigestax");  int times = bundle.getInt("times");  int rcount = bundle.getInt("rcount");  limited.setText(String.valueOf(bigestax));  scount.setText(String.valueOf(times));  aa.setText(String.valueOf(rcount));  }  //轉換角度標準值成Float  String a = limited.getText().toString();  float b = Float.parseFloat(a);  //轉換成功次數值成Float  String c = scount.getText().toString();  int d = Integer.parseInt(c);  //陀螺儀角度  if (sensorEvent.sensor.getType() == Sensor.TYPE\_GYROSCOPE) {  if (timestamp != 0) {  final float dT = (sensorEvent.timestamp - timestamp) \* NS2S;  angle[0] += sensorEvent.values[0] \* dT;  angle[1] += sensorEvent.values[1] \* dT;  angle[2] += sensorEvent.values[2] \* dT;  float anglex = (float) Math.toDegrees(angle[0]);  float angley = (float) Math.toDegrees(angle[1]);  float anglez = (float) Math.toDegrees(angle[2]);  tshow.setText("x : " + anglex + "\n" + "y : " + angley + "\n" + "z : " + anglez);  //如果角度大於角度標準值，開始執行程式，當角度小於角度標準值，跳至tmp.class並傳送角度值  if (anglex > b) {  new Handler().postDelayed(new Runnable() {  @Override  public void run() {  float late = (float) Math.toDegrees(angle[0]);  if (late + 0.1 < anglex) {  int count;  count = Integer.parseInt(aa.getText().toString());  count = count +1;  Intent tt = new Intent(test.this, tmp.class);  tt.putExtra("count", count);  tt.putExtra("tbigest", late);  tt.putExtra("bigestax",b);  tt.putExtra("times",d);  long rtime = Long.parseLong(spentTime.toString());  tt.putExtra("spentTime", rtime);  startActivity(tt);  finish();  }  }  }, 1);  }  //角度超過90跳回主頁面 \*並結算\*  if (anglex > 90) {  finish();  }  }  timestamp = sensorEvent.timestamp;  }  }  @Override  public void onAccuracyChanged(Sensor sensor, int i) {  }  } |

|  |
| --- |
| Record |
| package com.example.axix;  import android.content.Intent;  import android.database.Cursor;  import android.os.Bundle;  import android.view.View;  import android.widget.Button;  import android.widget.EditText;  import android.widget.TextView;  import android.widget.Toast;  import androidx.appcompat.app.AlertDialog;  import androidx.appcompat.app.AppCompatActivity;  public class record extends AppCompatActivity {  //SQL  EditText id, name, old, wid, mw, stime;  Button insert, update, delete, view;  DBHelper DB;  private TextView timeset;  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_record);  //接收值  timeset = (TextView)findViewById(R.id.textView3);  stime = (EditText)findViewById(R.id.stime);  Bundle re = getIntent().getExtras();  if (re != null) {  String bigestax = re.getString("ads");  String times = re.getString("ads");  timeset.setText(bigestax);  stime.setText(times);  }  //SQL  id = findViewById(R.id.id);  name = findViewById(R.id.name);  old = findViewById(R.id.old);  wid = findViewById(R.id.wid);  mw = findViewById(R.id.mw);  stime = findViewById(R.id.stime);  insert = findViewById(R.id.btnInsert);  update = findViewById(R.id.btnUpdate);  delete = findViewById(R.id.btnDelete);  view = findViewById(R.id.btnView);  DB = new DBHelper(this);  insert.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  String idTXT = id.getText().toString();  String nameTXT = name.getText().toString();  String oldTXT = old.getText().toString();  String widTXT = wid.getText().toString();  String mwTXT = mw.getText().toString();  String stimeTXT = stime.getText().toString();  Boolean checkinsertdata = DB.insertusertdata(idTXT, nameTXT, oldTXT, widTXT, mwTXT, stimeTXT);  if (checkinsertdata==true)  Toast.makeText(record.this, "新增資料！", Toast.LENGTH\_SHORT).show();  else  Toast.makeText(record.this, "新增失敗？", Toast.LENGTH\_SHORT).show();  }  });  update.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  String idTXT = id.getText().toString();  String nameTXT = name.getText().toString();  String oldTXT = old.getText().toString();  String widTXT = wid.getText().toString();  String mwTXT = mw.getText().toString();  String stimeTXT = stime.getText().toString();  Boolean checkinsertdata = DB.updateusertdata(idTXT, nameTXT, oldTXT, widTXT, mwTXT, stimeTXT);  if (checkinsertdata==true)  Toast.makeText(record.this, "資料更新成功！", Toast.LENGTH\_SHORT).show();  else  Toast.makeText(record.this, "資料更新失敗？", Toast.LENGTH\_SHORT).show();  }  });  delete.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  String idTXT = id.getText().toString();  Boolean checkinsertdata = DB.delectdata(idTXT);  if (checkinsertdata==true)  Toast.makeText(record.this, "資料刪除成功！", Toast.LENGTH\_SHORT).show();  else  Toast.makeText(record.this, "資料刪除失敗？", Toast.LENGTH\_SHORT).show();  }  });  view.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  Cursor res = DB.getdata();  if (res.getCount()==0){  Toast.makeText(record.this, "查無資料", Toast.LENGTH\_SHORT).show();  return;  }  StringBuffer buffer = new StringBuffer();  while (res.moveToNext()) {  buffer.append("id :"+res.getString(0)+"\n");  buffer.append("姓名 :"+res.getString(1)+"\n");  buffer.append("年齡 :"+res.getString(2)+"\n");  buffer.append("體重 :"+res.getString(3)+"\n");  buffer.append("性別 :"+res.getString(4)+"\n");  buffer.append("測驗時間 :"+res.getString(5)+"\n\n");  }  AlertDialog.Builder builder = new AlertDialog.Builder(record.this);  builder.setCancelable(true);  builder.setTitle("紀錄");  builder.setMessage(buffer.toString());  builder.show();  }  });  }  } |

|  |
| --- |
| DBHelper |
| package com.example.axix;  import android.content.ContentValues;  import android.content.Context;  import android.database.Cursor;  import android.database.sqlite.SQLiteDatabase;  import android.database.sqlite.SQLiteOpenHelper;  public class DBHelper extends SQLiteOpenHelper{  public DBHelper(Context context) {  super(context, "Userdata.db", null, 1);  }  @Override  public void onCreate(SQLiteDatabase DB) {  DB.execSQL("create Table Userdetails(id TXET primary key, name TEXT, old TEXT, wid TEXT, mw TEXT, stime TEXT)");  }  @Override  public void onUpgrade(SQLiteDatabase DB, int i, int i1) {  DB.execSQL("drop Table if exists Userdetails");  }  public Boolean insertusertdata(String id, String name, String old, String wid, String mw, String stime)  {  SQLiteDatabase DB = this.getWritableDatabase();  ContentValues contentValues = new ContentValues();  contentValues.put("id", id);  contentValues.put("name", name);  contentValues.put("old", old);  contentValues.put("wid", wid);  contentValues.put("mw", mw);  contentValues.put("stime", stime);  long result=DB.insert("Userdetails",null,contentValues);  if (result==-1){  return false;  }else{  return true;  }  }  public Boolean updateusertdata(String id, String name, String old, String wid, String mw, String stime)  {  SQLiteDatabase DB = this.getWritableDatabase();  ContentValues contentValues = new ContentValues();  contentValues.put("name", name);  contentValues.put("old", old);  contentValues.put("wid", wid);  contentValues.put("mw", mw);  contentValues.put("stime", stime);  Cursor cursor = DB.rawQuery("Select \* from Userdetails where id = ?", new String[]{id});  if (cursor.getCount() > 0) {  long result = DB.update("Userdetails", contentValues, "id=?", new String[]{id});  if (result == -1) {  return false;  } else {  return true;  }  }else {  return false;  }}  public Boolean delectdata(String id)  {  SQLiteDatabase DB = this.getWritableDatabase();  Cursor cursor = DB.rawQuery("Select \* from Userdetails where id = ?", new String[]{id});  if (cursor.getCount() > 0) {  long result = DB.delete("Userdetails", "id=?", new String[]{id});  if (result == -1) {  return false;  } else {  return true;  }  }else {  return false;  }  }  public Cursor getdata()  {  SQLiteDatabase DB = this.getWritableDatabase();  Cursor cursor = DB.rawQuery("Select \* from Userdetails", null);  return cursor;  }  } |

|  |
| --- |
| ChartLinerManager |
| package com.example.axix;  import com.github.mikephil.charting.charts.LineChart;  import com.github.mikephil.charting.components.AxisBase;  import com.github.mikephil.charting.components.Description;  import com.github.mikephil.charting.components.Legend;  import com.github.mikephil.charting.components.LimitLine;  import com.github.mikephil.charting.components.XAxis;  import com.github.mikephil.charting.components.YAxis;  import com.github.mikephil.charting.data.Entry;  import com.github.mikephil.charting.data.LineData;  import com.github.mikephil.charting.data.LineDataSet;  import com.github.mikephil.charting.formatter.IAxisValueFormatter;  import com.github.mikephil.charting.interfaces.datasets.ILineDataSet;  import java.text.SimpleDateFormat;  import java.util.ArrayList;  import java.util.List;  public class ChartLinerManager {  private LineChart lineChart;  private YAxis leftAxis;  private YAxis rightAxis;  private XAxis xAxis;  private LineData lineData;  private LineDataSet lineDataSet;  private List<ILineDataSet> lineDataSets = new ArrayList<>();  private SimpleDateFormat df = new SimpleDateFormat("HH:mm:ss");//設置日期格式  private List<String> timeList = new ArrayList<>(); //存儲x軸的時間  //折線圖  //多條曲線  public ChartLinerManager(LineChart mLineChart, String name, int color) {  this.lineChart = mLineChart;  leftAxis = lineChart.getAxisLeft();  rightAxis = lineChart.getAxisRight();  xAxis = lineChart.getXAxis();  initLineChart();  initLineDataSet(name, color);  }  //折線圖初始化  private void initLineChart() {  lineChart.setDrawGridBackground(false);  //顯示邊界  lineChart.setDrawBorders(true);  //折線圖例 標籤 設置  Legend legend = lineChart.getLegend();  legend.setForm(Legend.LegendForm.LINE);  legend.setTextSize(11f);  //顯示位置  legend.setVerticalAlignment(Legend.LegendVerticalAlignment.BOTTOM);  legend.setHorizontalAlignment(Legend.LegendHorizontalAlignment.LEFT);  legend.setOrientation(Legend.LegendOrientation.HORIZONTAL);  legend.setDrawInside(false);  //X軸設置顯示位置在底部  xAxis.setPosition(XAxis.XAxisPosition.BOTTOM);  xAxis.setGranularity(1f);  xAxis.setLabelCount(12);  xAxis.setValueFormatter(new IAxisValueFormatter() {  @Override  public String getFormattedValue(float value, AxisBase axis) {  return timeList.get((int) value % timeList.size());  }  });  //保證Y軸從0開始，不然會上移一點  leftAxis.setAxisMinimum(0f);  rightAxis.setAxisMinimum(0f);  }  //初始化折線  private void initLineDataSet(String name, int color) {  lineDataSet = new LineDataSet(null, name);  lineDataSet.setLineWidth(1.5f);  lineDataSet.setCircleRadius(1.5f);  lineDataSet.setColor(color);  lineDataSet.setCircleColor(color);  lineDataSet.setHighLightColor(color);  //設置曲線填充  lineDataSet.setDrawFilled(true);  lineDataSet.setAxisDependency(YAxis.AxisDependency.LEFT);  lineDataSet.setValueTextSize(10f);  lineDataSet.setMode(LineDataSet.Mode.CUBIC\_BEZIER);  //添加一個空的 LineData  lineData = new LineData();  lineChart.setData(lineData);  lineChart.invalidate();  }  //動態添加數據  public void addEntry(List<Float> number) {  if (lineDataSet.getEntryCount() == 0) {  lineData.addDataSet(lineDataSet);  }  lineChart.setData(lineData);  timeList.add(df.format(System.currentTimeMillis()));  for (int i = 0; i < number.size(); i++) {  Entry entry = new Entry(lineDataSet.getEntryCount(), number.get(i));  lineData.addEntry(entry, i);  lineData.notifyDataChanged();  lineChart.setVisibleXRangeMaximum(10);  lineChart.notifyDataSetChanged();  lineChart.moveViewToX(lineData.getEntryCount() - 5);  }  }  //設置Y軸值  public void setYAxis(float max, float min, int labelCount) {  if (max < min) {  return;  }  leftAxis.setAxisMaximum(max);  leftAxis.setAxisMinimum(min);  leftAxis.setLabelCount(labelCount, false);  rightAxis.setAxisMaximum(max);  rightAxis.setAxisMinimum(min);  rightAxis.setLabelCount(labelCount, false);  lineChart.invalidate();  }  //設置高限制線  public void setHightLimitLine(float high, String name, int color) {  if (name == null) {  name = "高限制線";  }  LimitLine hightLimit = new LimitLine(high, name);  hightLimit.setLineWidth(4f);  hightLimit.setTextSize(10f);  hightLimit.setLineColor(color);  hightLimit.setTextColor(color);  leftAxis.addLimitLine(hightLimit);  lineChart.invalidate();  }  //設置低限制線  public void setLowLimitLine(int low, String name) {  if (name == null) {  name = "低限制線";  }  LimitLine hightLimit = new LimitLine(low, name);  hightLimit.setLineWidth(4f);  hightLimit.setTextSize(10f);  leftAxis.addLimitLine(hightLimit);  lineChart.invalidate();  }  //設置描述信息  public void setDescription(String str) {  Description description = new Description();  description.setText(str);  lineChart.setDescription(description);  lineChart.invalidate();  }  } |

|  |
| --- |
| Firsttry |
| package com.example.axix;  import androidx.appcompat.app.AppCompatActivity;  import android.content.Intent;  import android.hardware.Sensor;  import android.hardware.SensorEvent;  import android.hardware.SensorEventListener;  import android.hardware.SensorManager;  import android.os.Bundle;  import android.os.Handler;  import android.os.Parcelable;  import android.view.View;  import android.widget.Button;  import android.widget.EditText;  import android.widget.TextView;  import java.util.ArrayList;  public class firsttry extends AppCompatActivity implements SensorEventListener {  //陀螺儀  private TextView show;  private SensorManager mSensorManager;  private Sensor macce;  private Sensor mgryo;  private Sensor mmagn;  private static final float NS2S = 1.0f / 1000000000.0f;  private float angle[] = new float[3];  //時間  private float timestamp;  //TextView最大值  private TextView bigest;  //時間  private Long startTime;  //延遲  private Handler handler = new Handler();  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_firsttry);  //取得目前時間  startTime = System.currentTimeMillis();  //設定定時要執行的方法  handler.removeCallbacks(updateTimer);  //設定Delay的時間  handler.postDelayed(updateTimer, 1000);  //賦予R.id  bigest = (TextView) findViewById(R.id.bigest);  show = (TextView) findViewById(R.id.showTextView);  mSensorManager = (SensorManager) getSystemService(SENSOR\_SERVICE);  macce = mSensorManager.getDefaultSensor(Sensor.TYPE\_ACCELEROMETER);  mgryo = mSensorManager.getDefaultSensor(Sensor.TYPE\_GYROSCOPE);  mmagn = mSensorManager.getDefaultSensor(Sensor.TYPE\_MAGNETIC\_FIELD);  }  //計時器  public Runnable updateTimer = new Runnable() {  public void run() {  final TextView time = (TextView) findViewById(R.id.timer);  Long spentTime = System.currentTimeMillis() - startTime;  //計算目前已過分鐘數  Long minius = (spentTime/1000)/60;  //計算目前已過秒數  Long seconds = (spentTime/1000) % 60;  Long mm = spentTime%1000;  time.setText(minius+":"+seconds+"."+mm);  handler.postDelayed(this, 1000);  }  };  //延遲設定  protected void onResume() {  super.onResume();  mSensorManager.registerListener(this, macce, SensorManager.SENSOR\_DELAY\_FASTEST);  mSensorManager.registerListener(this, mgryo, SensorManager.SENSOR\_DELAY\_FASTEST);  mSensorManager.registerListener(this, mmagn, SensorManager.SENSOR\_DELAY\_FASTEST);  }  //暫停程式時停止執行  protected void onPause() {  super.onPause();  mSensorManager.unregisterListener(this);  }  //輸出值  @Override  public void onSensorChanged(SensorEvent event) {  //陀螺儀角度  if (event.sensor.getType() == Sensor.TYPE\_GYROSCOPE) {  if (timestamp != 0) {  final float dT = (event.timestamp - timestamp) \* NS2S;  angle[0] += event.values[0] \* dT;  angle[1] += event.values[1] \* dT;  angle[2] += event.values[2] \* dT;  float anglex = (float) Math.toDegrees(angle[0]);  float angley = (float) Math.toDegrees(angle[1]);  float anglez = (float) Math.toDegrees(angle[2]);  show.setText("x : " + anglex + "\n" + "y : " + angley + "\n" + "z : " + anglez);  //次數標準值  new Handler().postDelayed(new Runnable() {  @Override  public void run() {  float late = (float) Math.toDegrees(angle[0]);  int p = 0;  int q = 0;  if (late > anglex)  {  bigest.setText(""+late);  }  if (late+0.1 < anglex)  {  bigest.setText(""+late);  Intent bt = new Intent(firsttry.this, MainActivity.class);  bt.putExtra("bigestax",late);  bt.putExtra("times", p);  bt.putExtra("fcount", q);  startActivity(bt);  finish();  }  if (anglex > 90)  {  finish();  Intent bt = new Intent(firsttry.this, MainActivity.class);  bt.putExtra("bigest","請重新測驗");  startActivity(bt);  }  }  }, 1);  }  timestamp = event.timestamp;  }  }  @Override  public void onAccuracyChanged(Sensor sensor, int i) {  }  } |

第十章、測試模型

10-1 測試計畫

　　此系統須為android 4.0以上的智慧型手機。確認每個功能是否正常執行。程式完成後，用城APK檔，到手機執行程式是否能正常執行。

10-2 測試個案與測試結果資料

本系統分別透過『HTC』、『MI』進行測試

|  |  |  |  |
| --- | --- | --- | --- |
| 測試編號 | Test01 | 測試人員 | 張勝崴 |
| 測試目標 | | 角度是否正常運作測試 | |
| 測試作業 | | | |
| 1. 進入APP 2. 測試角度是否正常顯示 3. 測試角度是否正常 | | | |
| 測試結果 | | 成功 | |

|  |  |  |  |
| --- | --- | --- | --- |
| 測試編號 | Test02 | 測試人員 | 江昱霆 |
| 測試目標 | | 按鈕是否正常運作測試 | |
| 測試作業 | | | |
| 1. 進入APP 2. 點選按鍵 3. 測試是否正常跳頁 | | | |
| 測試結果 | | 成功 | |

|  |  |  |  |
| --- | --- | --- | --- |
| 測試編號 | Test03 | 測試人員 | 康博閎 |
| 測試目標 | | 測試業面測試 | |
| 測試作業 | | | |
| 1. 進入APP 2. 進入測試頁面，開始測試 3. 測試是否成功有測試結果 | | | |
| 測試結果 | | 成功 | |

|  |  |  |  |
| --- | --- | --- | --- |
| 測試編號 | Test04 | 測試人員 | 郭金樑 |
| 測試目標 | | 歷史紀錄測試 | |
| 測試作業 | | | |
| 1. 進入APP 2. 開始測試，取得測試結果 3. 點選檢視測試數據 4. 測試是否有歷史紀錄 | | | |
| 測試結果 | | 成功 | |

|  |  |  |  |
| --- | --- | --- | --- |
| 測試編號 | Test05 | 測試人員 | 夏聖捷 |
| 測試目標 | | APP是否正常運作 | |
| 測試作業 | | | |
| 1. 進入APP 2. 進行第一次測試，取得角度標準值 3. 進入測試頁面 4. 新增使用者資料 5. 查詢測試結果 | | | |
| 測試結果 | | 成功 | |

第十一章、操作手冊

|  |  |
| --- | --- |
| 下載APK檔 | |
|  | 找到APK檔，進行下載 |
|  | 點「安裝」完成下載 |

|  |  |
| --- | --- |
|  | 跳出「已安裝應用程式」，代表已經下載好。 |
|  | 到桌面，看到捷進，就可以進入APP。 |

第十二章、使用手冊

|  |  |
| --- | --- |
|  | 開啟APP，進入主畫面。 |
|  | 進入如何使用，查看如何操作。 |
|  | 進入首次測驗，取得測驗標準角度。 |
|  | 回到主頁，輸入測驗次數，點開始測驗。 |
|  | 進行測驗，取得測驗結果。 |
|  | 回到頁首，點紀錄結果，查看紀錄。 |
|  | 進到紀錄結果，填好資料，按新增資料，跳出文字，表示新增成功。 |
|  | 點查看紀錄，就可以看到所有資訊。 |
|  | 將新的資料打上，點更新資料，看到跳出文字，表示資料更新完成。 |
|  | 如果要刪除資料，點擊刪除資料，看到跳出文字，表示資料已經刪除。 |
|  | 刪除資料後，想確認是否刪除，可以點擊查看紀錄，會跳出查無資料。 |

第十三章、感想

N1076405 張勝崴

這次的專題我非常感謝老師的指導，還有調整我的心態，因為程式只有我在寫，所以時常壓力很大，也很常跟上班的公司請假，都是為了寫程式，雖然對於讀夜校以及半工半讀的我來說非常辛苦，但換個方式說，就是非常的充實，受益良多，憑自己從剛開始3月都不會寫程式到現在來比，我真的成長了非常多。

N1076406 江昱霆

這一次的專題我們的環境是android studio， 一開始我們都沒有程式的經驗，因為這一次專題讓我們對java這個區塊，有機會對它有了初步的認識， 由於關於我們程式開發相關的資料非常的少，再者程式的功力也還不足 ，程式一開始的開發可以說相當的艱辛，在我們不斷的努力和嘗試 ，組員辛苦的從網路上不斷的找相關知識，一步一腳印的從程式無到有，中間花費了非常多的時間摸索，所以當成果越來越完美的時候，心裏都感覺非常的有成就感，也是因為這樣從無到有，我們才有這一次專題的成果，經過這次的學習我們對程式介面的開發、程式的技巧 ，以及最後的整合都有了解與認識 ，這跟學校學到理論相當不同，專題是把所學的理論實現出來 ，當最後看著成果出來 成功將數據顯示成圖形化的那時候就非常的讓我們興奮，之後要上台報告自己目前的進度以及上台之前的準備 ，讓我們學習不

少如何將自己目前工作狀況表達給其他人 ，和如何讓一個team順利的工作 ，這是一般課程學習不到的！

N1076411 康博閎

1. 專題是包含了團隊合作、資料與知識的搜索、整合、吸收、運用、積極主動的態度與熱情以及團隊中每個人的經歷、能力、價值觀的表現、學習進步與共同磨合的過程。
2. 當初跟老師提出建議主題時，就發現資訊與醫療結合的主題在我所見所聞的專題主題中是非常少見的，所以我們一致同意此主題，但開始寫程式功能時才發現手機APP要做到關醫療測量器的困難程度跟我們當初大一大二學習的程式難度簡直是天差地遠，光一個陀螺儀的功能呈現、校準就可以耗費我們數月時間。
3. 但是我個人覺得這次的團隊合作最困難的是就讀夜間部的我們，各各都身處不同的行業，要擠出上課以外的時間討論專題的機會是少之又少，再加上今年年中的疫情爆發，工作量增加與大三大四的課業加重讓我們專題的進度較緩慢，好幾次討論都是犧牲大家睡覺時間趕進度，但也意識到調配利用時間的重要性。

有別於我們主題的特殊性，要寫出一個創新、跨領域、沒有非常近似的軟體可比照的手機APP是對我們是相當有挑戰性的。不過剛好可以透過這次專題讓我們可以真正了解我們擅長的與不熟悉的能力在哪裡。

N1076414 夏聖捷

心得:大學這唯一一次而且最重要的專題，一路下來，我們的主程式設計勝崴揹著相當大的壓力，我們只能主要協助幫他找他需要的資料以及後期測試，我知道這次專題最重要的就是程式，但我真心在程式這方面幫不上甚麼忙，只能儘量在這以外的地方幫忙。大家辛苦了，尤其主程式設計!

N1076448 郭金樑

這次的專題讓我體會到團隊合作與老師溝通的重要性，因為一個人做Word的時候真的很痛苦，卡住時沒有人幫忙出意見真的做不出任何東西，與老師meeting後，老師給了我突破性的意見，也有了隊友的幫忙後，許多的問題都被一一解決，Word也完成了。所以經過這個專題讓我了解也些事情需要團隊一起合作才能快速完成。

第十四章、參考資料

1.wall slides

<https://www.verywellfit.com/wall-slides-an-effective-quad-strengthening-exercise-2696607>

2.折線圖

<https://medium.com/@makkenasrinivasarao1/line-chart-implementation-with-mpandroidchart-af3dd11804a7>

3.SQLite

<https://www.google.com.tw/amp/s/www.stechies.com/sqlite-database-update-android/>

4.傳感器

<https://developer.android.com/reference/android/hardware/SensorEvent>

5.換頁資料傳輸

<https://stackoverflow.com/questions/14876273/simple-example-for-intent-and-bundle>

6.學習三維角度

<https://silverwind1982.pixnet.net/blog/post/258069682>

7.四元數

https://zh.m.wikipedia.org/zh-tw/四元數

附錄

審查評審意見之修正情形

|  |  |
| --- | --- |
| 評審建議事項 | 修正情形 |
|  |  |