

OOP Final Project Group 1 Report

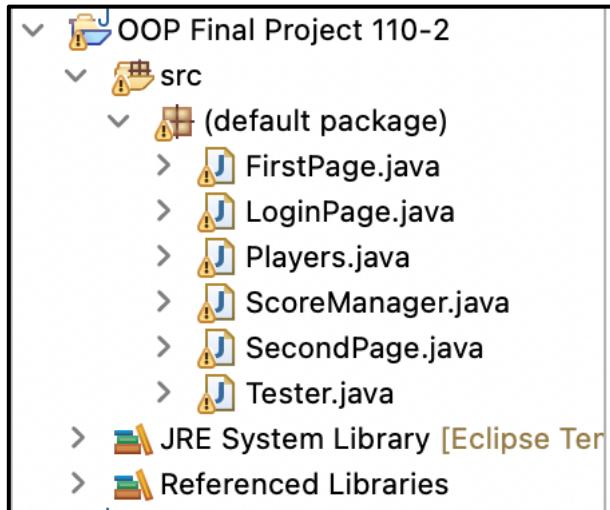
組員：110306047 吳堃豪 110306068 張奕奇 110306002 許馨文 109306072 邱郁宸

背景與動機：由於我們四個人都是對排球運動有極高熱情的人，有三位在系隊上打球另一位則是球經，在練習及比賽的過程我們發現紀錄排球是一件繁瑣冗長的事情，於是想要開發一個小型的軟體，方便球經或是教練紀錄球員的表現等。

使用到的工具：Java, SQL

系統架構：內部使用了六個 class，外加 sql 模組，分別為：

1. LoginPage
2. FirstPage
3. Players
4. ScoreManager
5. SecondPage
6. Tester



Class 及邏輯介紹

1. Players

此 class 乃為了產生 Players 的 Object，帶有四個 parameter 分別是姓名、背號、位置、以及性別，非常簡單明瞭。

```
public Players(String name, int number, String position, String sex) {  
    this.number = number;  
    this.position = position;  
    this.name = name;  
    this.sex = sex;  
}
```

2. Tester

不用多做贅述，主要是叫出 LoginPage 的畫面便結束。

3. LoginPage

此 class 用於呈現登入頁面，讓使用者可以登入及註冊帳

號密碼，註冊後的資料會全數儲存在 SQL 資料庫上面，

特別的是我們將密碼在儲存及驗證時都使用 SHA-256 加

密，若成功登入便會進入 FirstPage.



I. 關於加密部分

使用 java 內建的

MessageDigest 功能進行 SHA-

256 之加密過程，將加密後的

密碼上傳至 SQL，由於 SHA-

256 是一種雜湊加密函數，它

是難以被回推跟解碼的，故帳

號安全性可以得到相對的保

證，如下圖。

```
public String getSHA256StrJava(String str) {
    MessageDigest messageDigest;
    String encodeStr = "";
    try {
        messageDigest = MessageDigest.getInstance("SHA-256");
        messageDigest.update(str.getBytes("UTF-8"));
        encodeStr = byte2Hex(messageDigest.digest());
    } catch (NoSuchAlgorithmException e) {
        e.printStackTrace();
    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    }
    return encodeStr;
}

private String byte2Hex(byte[] bytes) {
    StringBuffer stringBuffer = new StringBuffer();
    String temp = null;
    for (int i = 0; i < bytes.length; i++) {
        temp = Integer.toHexString(bytes[i] & 0xFF);
        if (temp.length() == 1) {
            stringBuffer.append("0");
        }
        stringBuffer.append(temp);
    }
    return stringBuffer.toString();
}
```

account	password
123	a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e99...
456	b3a8e0e1f9ab1bfe3a36f231f676f78bb30a519d2b21e6c530...
liv0123	7a874734355315df05fa7ac8e580093c78a9f51b3862f9ef1e...

II. 使用 passwordfield

```
private JPasswordField passwordfield;  
private JButton enrollButton, loginButton;  
protected String loginname = "";  
private String query = "";
```

如圖，我們將 password 的輸入欄位改為 JPasswordField 的形式，使輸入時密碼欄位會如下所示：



一方面此種辦法可以保障使用者在輸入時的資安，亦可以使整個登入介面看起來更為真實、更為專業。

4. FirstPage

I. 主菜單頁面

一進來 FirstPage 後首先迎來的是使用 CardLayout 寫成的 Menu，讓使用者可以選擇創建一個新的記錄表，或是查看此資料之前的歷史紀錄，如右圖所示。



II. 歷史紀錄頁面

此頁會抓取 SQL 以前的紀錄，將資料全部列出，是非常基本的 SELECT 語法所構成。

The screenshot shows a window titled '自動化排球紀錄系統' (Automated Volleyball Record System). The main title is '123的歷史紀錄'. Below it, there is a timestamp '時間 2022/06/07 14:39:51' and a location '地點 政大五期排球場'. To the right, it displays team scores: 'A隊名稱: A隊得分 MIS: 25分' and 'B隊名稱: B隊得分 BA: 20分', with '結果 MIS勝'. A note below says '以下空白'. At the bottom right is a '返回' (Return) button.

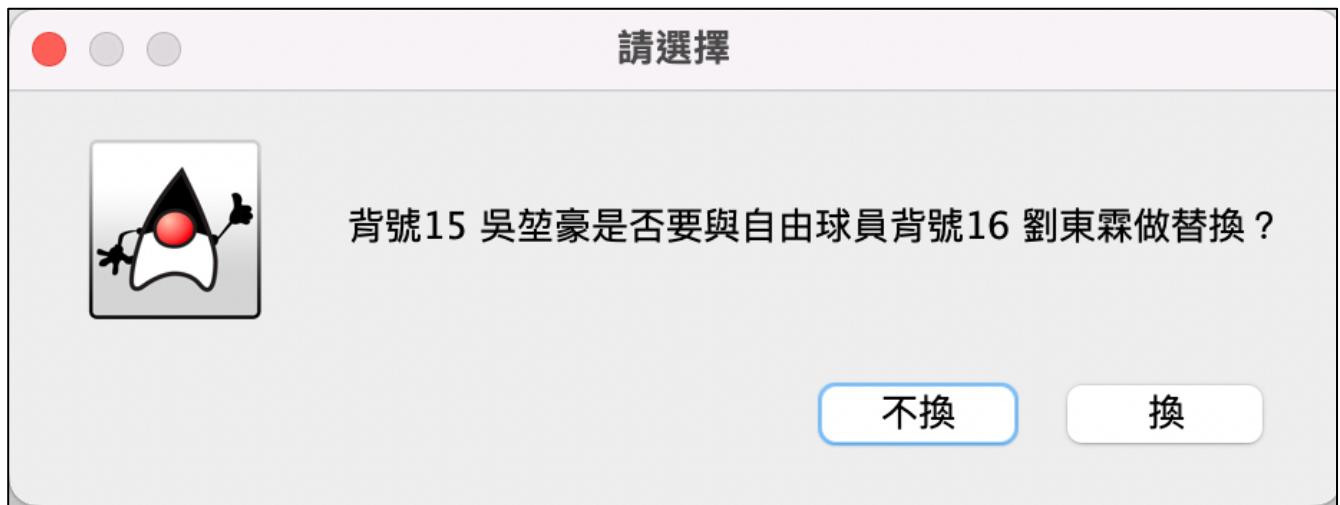
III. 紀錄頁面

The screenshot shows a window titled '自動化排球紀錄系統'. It has a text input field '請輸入您的名字' (Please enter your name) and a date/time input field '2022/06/07 14:46:00'. On the left, there are dropdown menus for '請輸入我方隊伍名稱' (Please enter our team name), '請輸入敵方隊伍名稱' (Please enter the opponent's team name), '比賽地點' (Competition location), and '比賽時間' (Competition time). On the right, there is a table with columns '姓名' (Name), '背號' (Back number), '位置' (Position), and '性別' (Gender). The table contains six rows, each with a dropdown menu for position and gender. Buttons for '開始' (Start) and '返回' (Return) are at the bottom right.

接下來來到整個專案最為重點的部分之一，也就是 key in 球員的資料以及性別位置。當初在構想將性別也納入選單的原因是因為可以配合米克斯盃（也就是男女混排的比賽）去記錄使用，左方的名字、隊伍名稱、敵方隊伍名稱、比賽地點時間都會在最後產生的報表中出現，時間的話是採用當下的時間，也就是使用 Java 內建的 DateTimeFormatter 加上 LocalDateTime.now()去做時間抓取，至於自由球員的部分會先詢問是否有需要添加自由球員，需要的話再進一步做輸入，如下圖所示。



輸入完自由球員的資料後，由於排球比賽通常會把攔中安排在前排，到後排時會替換成較為擅長接球的自由球員，因此系統會去偵測球員中是否有人為攔中，若有的話會詢問是否替換，如圖所示。



若選擇替換，在往後的比賽過程中將會自動偵測，若攔中轉到後排且發球輪結束，便會自動替換為選取的自由球員，反之自由球員到前排時也會自動切換成攔中。選完自由球員便是選擇先發球抑或

是後發球，此會決定待會的 Label 樣式以及整體輪轉表的形式。



5. ScoreManager

I. 說明作用

此 class 是待會進入到紀錄介面時，為了避免 SecondPage 過度冗長所新增的一個，專門用來處理數據計算的 class，其功能有

1. 判斷輪轉/發球邏輯
2. 判斷輸贏邏輯(包括 deuce)
3. 進行個別球員資料之紀錄（發球得分、攔網得分、攻擊得分、失誤失分）
4. 紀錄總體得分分佈（我方得分/敵方失誤導致我方得分/我方受迫性失誤失分）

II. 發球輪轉邏輯講解

排球規則乃由發球開始，發球員擊球過網進入對區，展開雙方來回對擊，直到球落地、出界或某隊未能合法將球擊回對區。排球比賽中，球隊贏得一球，即可得到一分。當接發球隊贏得一球時，該隊得到一分，並獲得發球權，球員順時鐘方向輪轉一個位置。也就是說「當我方失分後又再次得分獲得發球權」便要以順時針方向輪轉一個位置。

當我（吳堃豪）在撰寫這個部分時我一時之間不曉得該如何將這麼文字化的東西變成邏輯性、電腦可以解讀的事情。後來我採用 append 的方式，也就是建立一個 String，在敵方得分時將 String 後方附加一個”0”，我方得分時附加一個”1”，在每次按下按鈕紀錄時都去檢查此 String 的「倒數兩個數字」是否為”01”，因為若為 01，代表剛剛敵方得分完換我方得分，也就是前面所述的「當我方失分後又再次得分獲得發球權」，系統便會去執行輪

轉。

```
if (orderString.length() >= 2) {  
    if (orderString.substring(orderString.length() - 2).equals("01")) {  
        order = changeorder();  
    } else {  
        return;  
    }  
}
```

至於輪轉是如何進行，我寫了一個判斷，將右下角的球員原本在幾號位變成一個 int 變數(order)，如

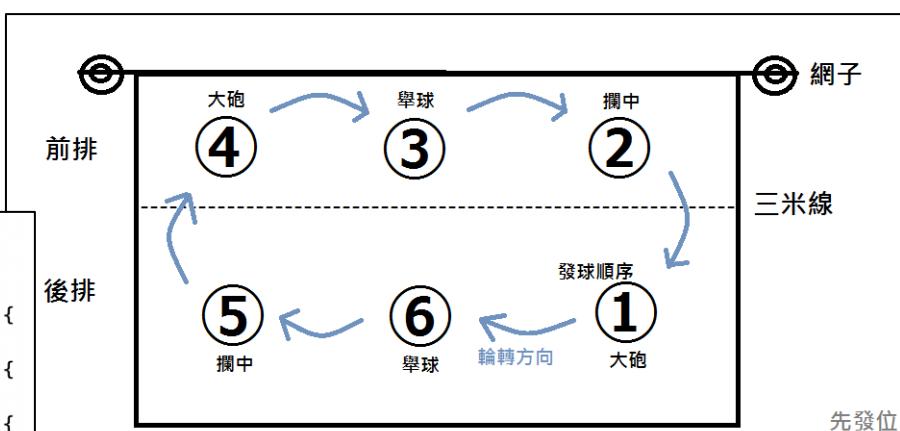
果原本是 1 號位，那麼輪轉過後就應該

是 2 號位（如右圖），依此類推的話，

便形成如右

的程式碼。

```
public int changeorder() {  
    if (order == 1) {  
        order = 2;  
    } else if (order == 2) {  
        order = 3;  
    } else if (order == 3) {  
        order = 4;  
    } else if (order == 4) {  
        order = 5;  
    } else if (order == 5) {  
        order = 6;  
    } else if (order == 6) {  
        order = 1;  
    }  
    return order;  
}
```



將 order 變為「新的輪轉形式右下角的人的原本的 n 號位」後，接著就是在 SecondPage 中使用 if else 去判斷現在的輪轉應該長什麼樣子，這個部分待會會說明。

6. SecondPage

接下來便是來到最為複雜但也最為重要的 SecondPage，全 class 長達 1700 多行，其功能便是紀錄數據，藉由按鈕的形式，系統會自動去計分、判斷勝負、進行輪轉、匯出報表(.csv)，可以說這個專案的精華都在此 class 當中。

4	name4	攔中	男	3	name3	舉球	男	2	name2	大砲	男
發球得分	攔網得分	攻擊得分	失誤失分	發球得分	攔網得分	攻擊得分	失誤失分	發球得分	攔網得分	攻擊得分	失誤失分
5	name5	大砲	男	6	name6	舉對	男	1	name1	攔中	男
發球得分	攔網得分	攻擊得分	失誤失分	發球得分	攔網得分	攻擊得分	失誤失分	發球得分	攔網得分	攻擊得分	失誤失分

上面為 SecondPage 的整體版面，上面是計分區，中間是發球顯示及我方受迫性失分/敵方失誤導致我方得分的顯示紀錄區，下面的六個部分分別為六位球員的基本資料，每位球員都有四個按鈕去紀錄其得分或失分。

I. 資料的傳遞

由於在資料的傳遞過程中我們發現資料在 class 之間傳遞會有遺漏跟無法傳遞的問題，於是我們將 SecondPage 的 constructor 加入前面 FirstPage 的 Players 的 ArrayList，不僅傳送球員的資料，也將隊伍名稱、地點、時間、記錄者等一起放在裡面傳送過去，詳細如下 excel 所示。

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13
name	player1.name	player2.name	player3.name	player4.name	player5.name	player6.name	teamname	enemyteamname	time	recorder's name	place	accountname	libro1.name	libro2.name
number	player1.num	player2.num	player3.num	player4.num	player5.num	player6.num	""	""	""	""	""	""	libro1.num	libro2.num
position	player1.position	player2.position	player3.position	player4.position	player5.position	player6.position	""	""	""	""	""	""	libro1.position	libro2.position
sex	player1.sex	player2.sex	player3.sex	player4.sex	player5.sex	player6.sex	""	""	""	""	""	""	libro1.sex	libro2.sex

II. 按鈕的 callback listener

```
public void actionPerformed(ActionEvent e) {
    scoreManager.addservespoint(1);
    serves.set(1, serves.get(1) + 1);
    score.setText(players.get(6).getname() + " : " + players.get(7).getname() + " = "
        + scoreManager.getAteamScore() + " : " + scoreManager.getBteamScore());
    winner = scoreManager.checkwin(scoreManager.getAteamScore(), scoreManager.getBteamScore());
    if (winner.equals("win")) {
        write();
        System.out.println("Output successfully");
    } else if (winner.equals("lose")) {
        write();
        System.out.println("Output successfully");
    } else if (winner.equals("null")) {
        scoreManager.checkorder();
        createdataJPanel();
    }
}
```

解釋在按下每一個 button 的時候會執行什麼指令，指令如下：

1. 若為得分則在總得分+1，得分球員個人的資料也會+1(serves 的 ArrayList, set(number1, +1 分))
2. 加分後重新載入計分板
3. 確認輪轉，重新繪製底下的輪轉圖
4. 判別勝負
5. 若比賽結束，進行輸出(write())

而一樣的東西，只是每次都要稍作修改的其他 23 個 buttons 都是如法炮製。

III. 按鈕的邏輯判定

若為敵方發球，我方很合理的不可能會有「發球得分的情況出現」；又或著是我方發球，只可能由發球者產生「發球得分」，其他人不可能會有發球得分的情況出現，因此，在這兩種情況發生時，我會將所有其他的發球得分按鈕關閉(setEnabled(false))，變成如下圖的形式；同理當自由球員在後排時也不可能攔網得分，同樣 setEnabled(false)。

4	4	大砲	男	3	3	大砲	男	2	2	大砲	男
發球得分	攔網得分	攻擊得分	失誤失分	發球得分	攔網得分	攻擊得分	失誤失分	發球得分	攔網得分	攻擊得分	失誤失分
5	5	大砲	男	6	6	大砲	男	1	1	大砲	男
發球得分	攔網得分	攻擊得分	失誤失分	發球得分	攔網得分	攻擊得分	失誤失分	發球得分	攔網得分	攻擊得分	失誤失分

IV. 輪轉邏輯以及重繪之操作

輪轉的編寫花了我們非常多時間和精力，一開始的方式是讓選手(playerpanel)站到場上後，在(非

```
numb1serve.setEnabled(false);
numb2serve.setEnabled(false);
numb3serve.setEnabled(false);
numb4serve.setEnabled(true);
numb5serve.setEnabled(false);
numb6serve.setEnabled(false);
```

我們發球&&得分)時，直接移動選手，但執行上非常困難，因為我們無法用程式去判定現在的輪轉位置，因此我們改以載體(p1~p6panel)的方式，讓場上先存在六個載體，在初始化的時候直接將選手放入載體中，在輪轉過程直接移動載體後再重繪整個場上的panel，免除判定選手位置的麻煩，要替換自由球員時也可以直接將載體內的選手做更換。



不妨使用 minecraft 中船的想法當作載體

V. 結束匯出之手法

在記錄完成直到有一方勝利時，系統會自動跳出檔案選擇的視窗，讓使用者自己選擇檔

名以及儲存位置，介面如右。我

使用 JFileChooser 來輸出.csv 檔

案，包括未打副檔名自己 append

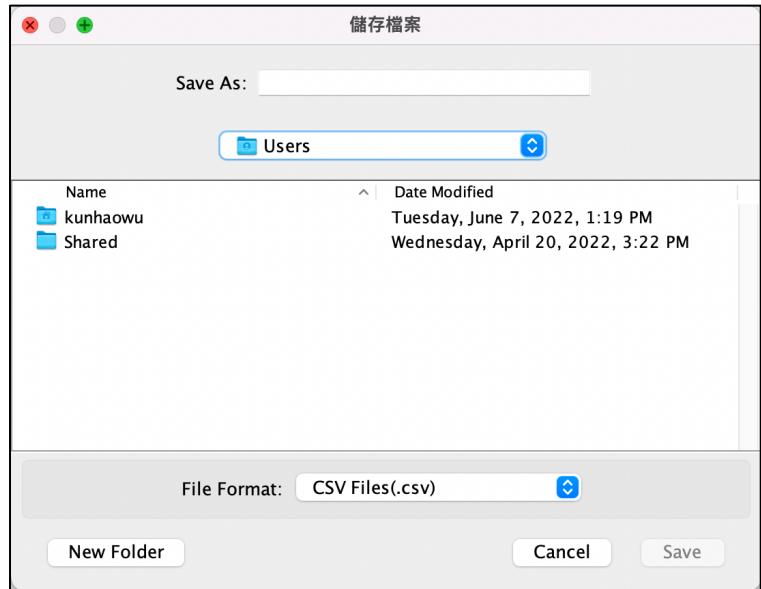
等等，以及使用","的分隔符號讓

其自己跑出如 excel 表格般的形

式，藉由非常簡單的逗號分隔來

產生一目瞭然，可以供應各種需

求的檔案。詳細跑出的形式請見下頁。



```
JFileChooser fs = new JFileChooser(new File(""));
fs.setDialogTitle("儲存檔案");
fs.setFileFilter(new FileFilter() {
    @Override
    public String getDescription() {
        // TODO Auto-generated method stub
        return "CSV Files(.csv)";
    }

    @Override
    public boolean accept(File f) {
        // TODO Auto-generated method stub
        return f.isDirectory() || f.getName().toLowerCase().endsWith(".csv");
    }
});
```

123

排球自動化紀錄系統報表							
地點: 政大五期排球場							
時間: 2022/06/07 14:39:51							
記錄者: 吳堃豪							
結果: MIS勝							
最終比數 MIS 25 : BA 20							
球員姓名	背號	位置	性別	發球得分	攔網得分	攻擊得分	失誤失分
1	1	大砲	男	1	1	0	0
2	2	大砲	男	1	2	2	-2
3	3	大砲	男	0	0	0	0
4	4	大砲	男	0	0	0	0
5	5	大砲	男	0	0	0	0
6	6	大砲	男	0	0	0	-1
對方失誤導致我方得分	20						

輸出後同時會將資料上傳至 SQL 雲端資料庫，以 accountname 作為分別，讓下次登入者可以查詢自己的歷史紀錄（如下圖）

顯示第 0 - 0 列 (總計 1 筆, 查詢用了 0.0001 秒。)
SELECT * FROM `records`
<input type="checkbox"/> 效能分析 [行內編輯] [編輯] [SQL 語句分析] [建立 PHP 程式碼] [重新整理]
<input type="checkbox"/> 全部顯示 資料列數： 25 <input type="button" value="↑"/> 篩選資料列: 搜尋此資料表
+ 選項
accountname time place teamAname teamPoints teamBname teamBpoints consequence
123 2022/06/07 14:39:51 政大五期排球場 MIS 25 BA 20 MIS勝

組內分工與組內互評：

吳堃豪(組長)：程式架構與後端、建構 SQL、期末報告 Word 檔案、分派組內分工與監督死線

張奕奇：負責撰寫邏輯運算以及 GUI 的重繪、SQL 命令與連線處理、期末報告的說明書以及打包

許馨文：負責 GUI 設計、登入頁面、FirstPage/SecondPage 的 GUI Design、期末報告的簡報

邱郁宸：負責 GUI 設計、CardLayout、SQL 架構與命令句、期末報告的簡報

互評：無論是一開始的 idea、整體架構、成堆的 bug 要 de、困難的 GUI design、SQL 的建置聯絡，

每個人都貢獻良多，因次無論是什麼方面我們都給互相滿分的成績，很慶幸可以擁有彼此一同擔任組員，這次的經驗獲益良多且難能可貴。

心得

吳堃豪

首先這個我得承認這個專案一點都不花俏，坦白說蠻平鋪直敘的，但是這個專案中牽涉到的邏輯以及 GUI 真的花我們非常多的時間，雖然不像其他組有什麼花俏的模組，但是我們用心做出我們自己需要、而且想要的軟體與樣式，我覺得這就讓我獲益良多以及更熱愛程式。可以說，我們將這學年的程式設計課教過的東西幾乎都發揮得淋漓盡致，每個東西都造就了他最大的價值，再次感謝我的組員與我合作的非常愉快，過程中真的獲益良多，大功告成時的成就感也絕非一般可以比擬的！

張奕奇

這次加入這組蠻開心的，在這組裡不會有那種做團體報告的壓力，大家的分工蠻明確的，而且不會想說話的時候不敢說，也不需要扛過多的東西，這次我負責的基本上就是一些 debug 還有輪轉、幫忙寫資料庫邏輯的一些工作，感謝吳堃豪寫了大部分的程式，雖然我跟他寫程式的邏輯差很多導致我要 debug 的時候都要一直問他這是什麼意思，但他扛了大概 20000 字，牛。也謝謝邱郁宸提供了良好的經驗，許馨文提供了良好的場地，他們兩個一起提供了良好的簡報，我想這應該是我第一次做團體報告有那麼快活的氣氛，如果吳堃豪不會嫌棄我們太沒用那下次肯定還要一起合作。

許馨文

由於課程內容的複雜度增加，這學期的專案內容跟上學期比起來完全不是同個級別，上手也困難許多。主要很感謝我的組員們願意在我困惑時為我一一解惑，大家都各自負責不同的領域讓這個專案變得多采多姿。先層層地去了解程式碼的運作後，再對其做美編或更深的邏輯推算就不是件難事。雖然中間一度查 google 查到快崩潰還跳出「java 從入門到放棄」的字眼，但一想到我的組員比我更爆肝就硬著頭皮的寫下去，嗚嗚。

邱郁宸

當我在設計 GUI 的時候，遇到兩個較難解決的問題是，資料在 JAVA 及 SQL 之間互轉會產生亂碼，導致排版不符合預期，最後也在大家共同找資料以及研究下，順利將該問題解決。另外，為了讓畫面能夠同時選擇「歷史紀錄」以及「創建新的資料庫」，我們使用了 cardlayout 讓畫面更貼近我們想像的樣貌，在使用 cardlayout 的時候因為較不熟悉，所以花了一些時間在這上面。最後則是非常感謝，其他組員對我的幫助，讓我就算遇到困難也可以順利解決。所以我給他滿分。