行動應用程式開發專題

書面報告

野球紀錄

組員：董育汝、曾玉鳳、劉永貴

中華民國一百零捌年陸月

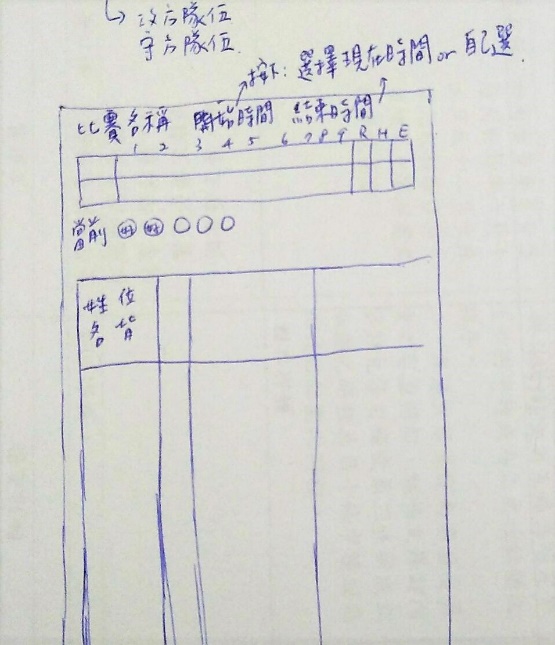
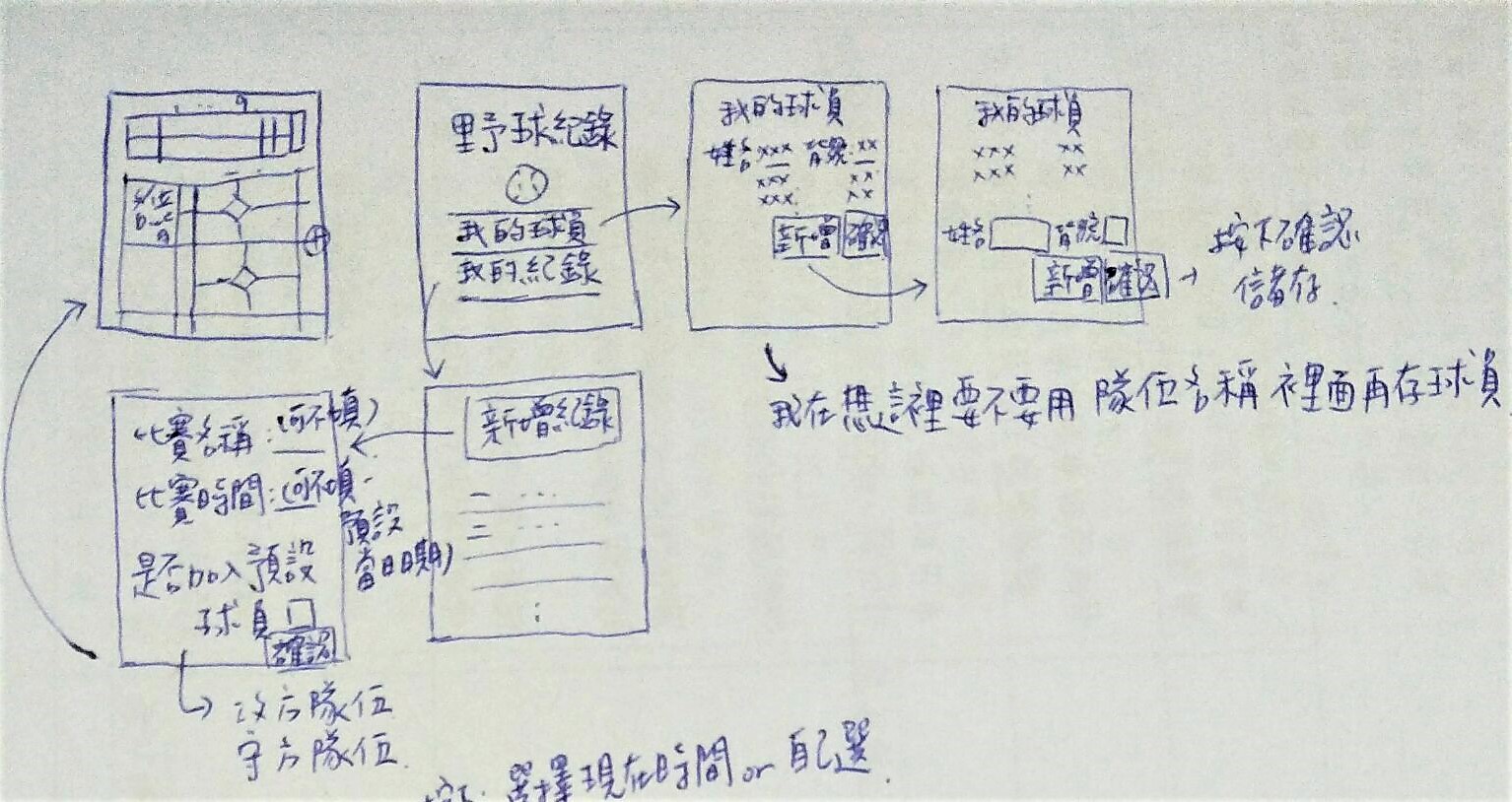
一、內容 (what- function)

現有棒球紀錄的應用程式較少，大多都是透過文字敘述來選擇、記錄，以每個棒次為單位紀錄，無法記錄到之後的跑壘狀態或是其他狀況，較不精確、詳細，也很不彈性。

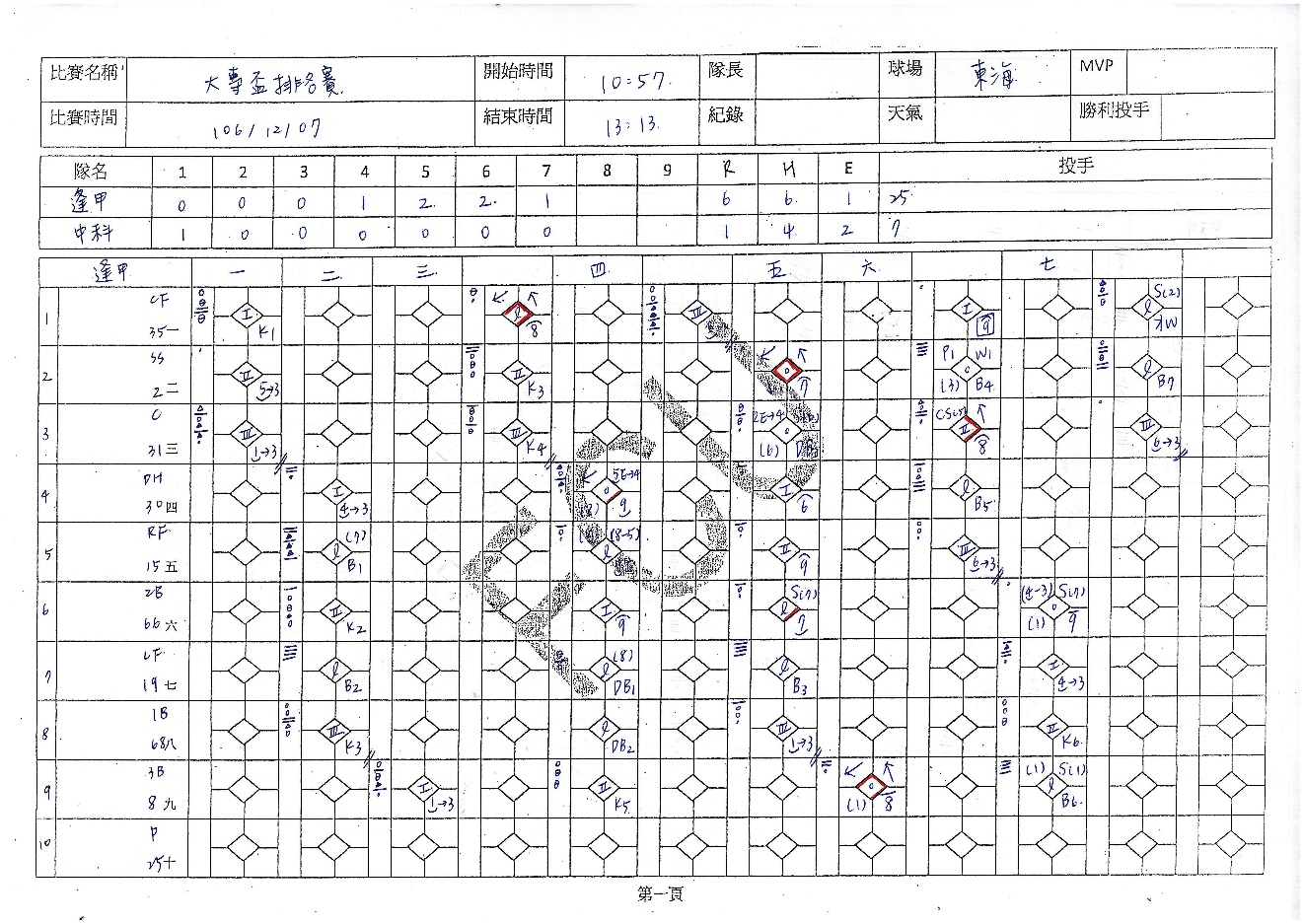
我們將原本的紙本紀錄方式，改良製作成手機應用程式，透過直觀的原本該記錄在哪一格就點選該格做選擇，大部分資料手動控制，較能符合比賽中許多的狀況。並且能夠輸入自己球隊的隊伍名稱及隊員，透過排序球員讓新增比賽時能自動加入前9位球員，也省去了每比賽一次就要全部重新輸入或紙本寫上球員名稱的困擾。

二、設計 (how- design)

(圖一)為設計原稿，首頁點選後進入我的球員、我的紀錄。



(圖一)介面設計圖



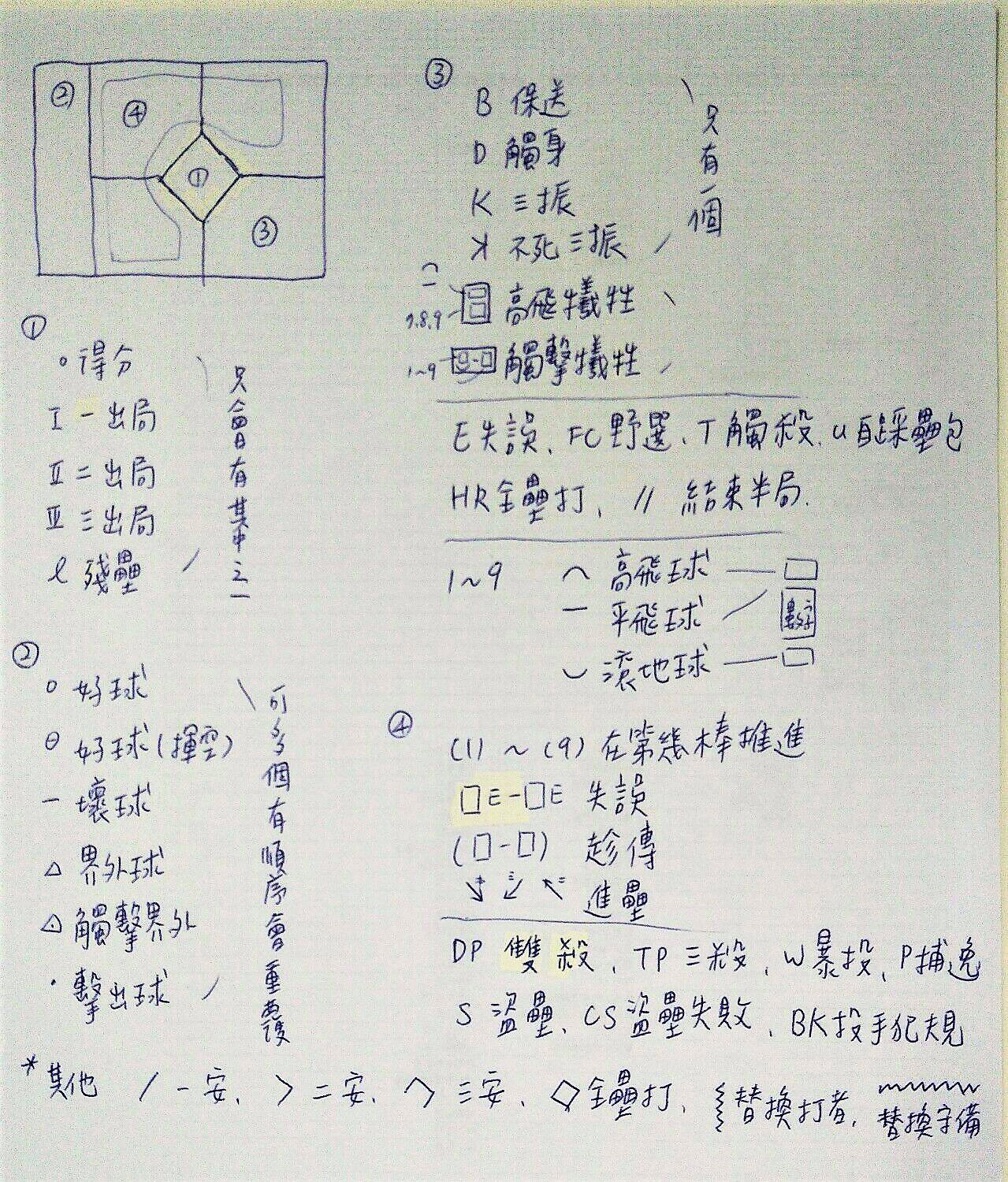
選手名稱

(圖二) 紙本棒球紀錄表

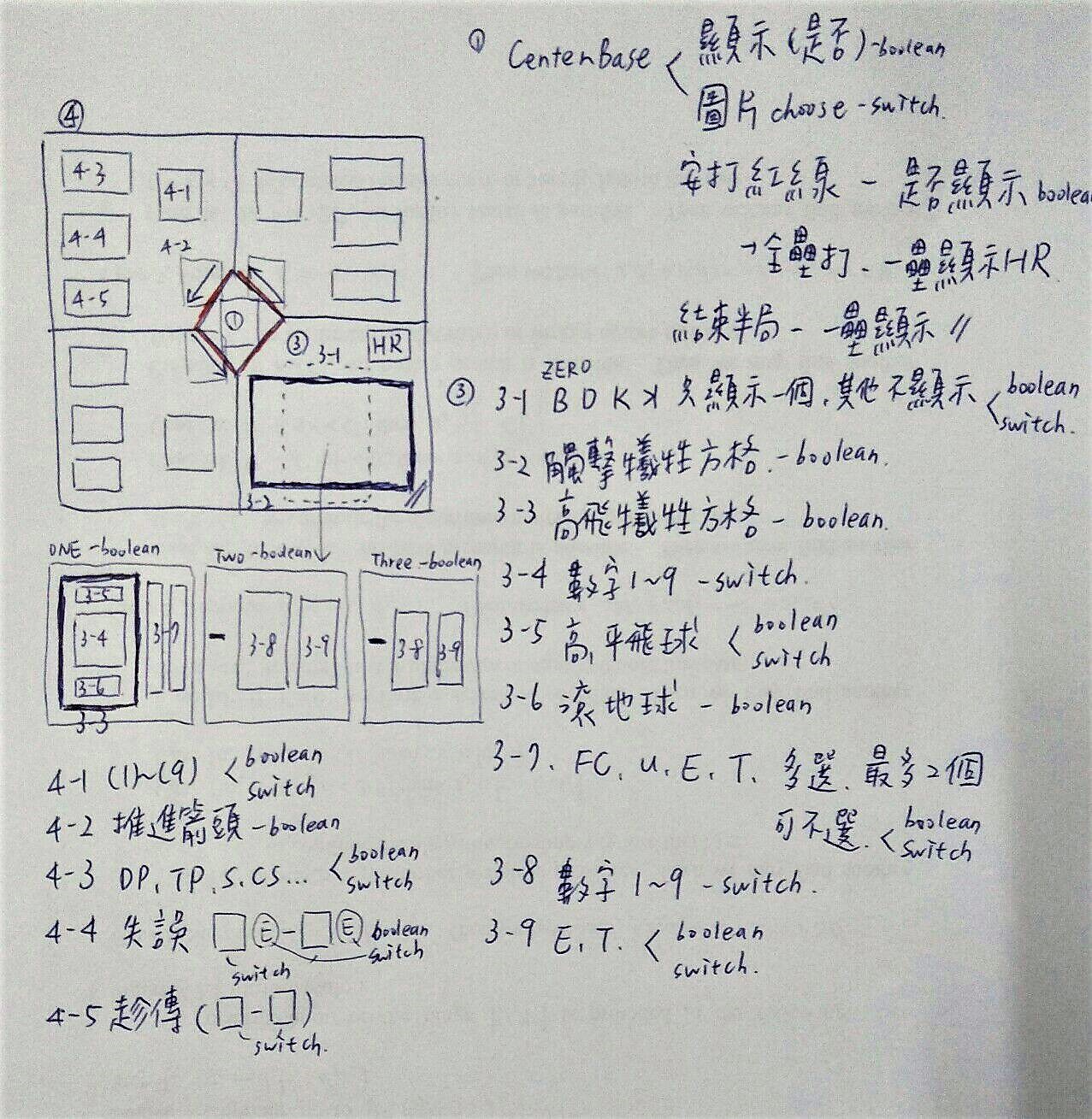
利用(圖二)棒球紀錄方法，製作出電子版紀錄。

紅色框框中設計為固定顯示。

藍色框框中內容為滑動顯示，透過滑動顯示所有紀錄。



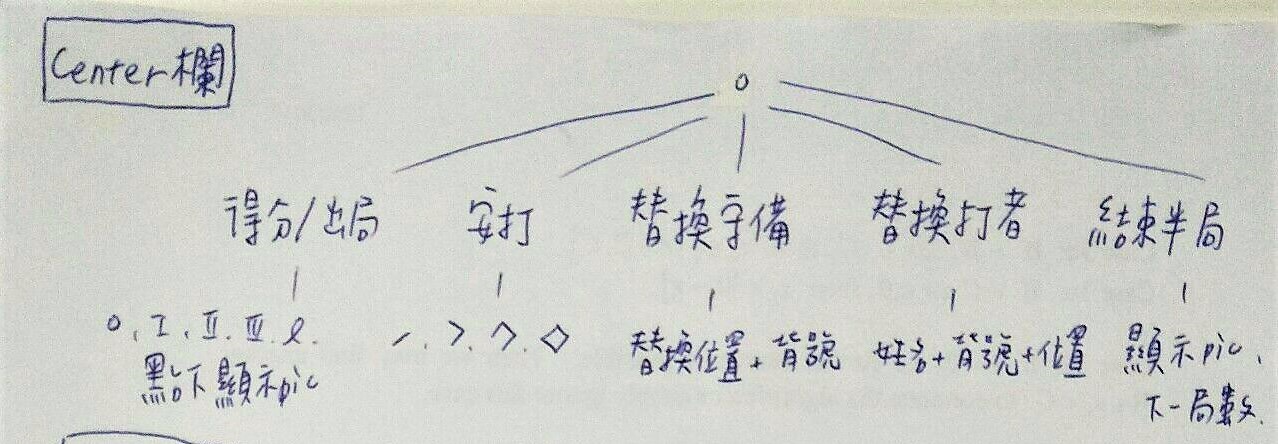
(圖三) 棒球常用紀錄符號



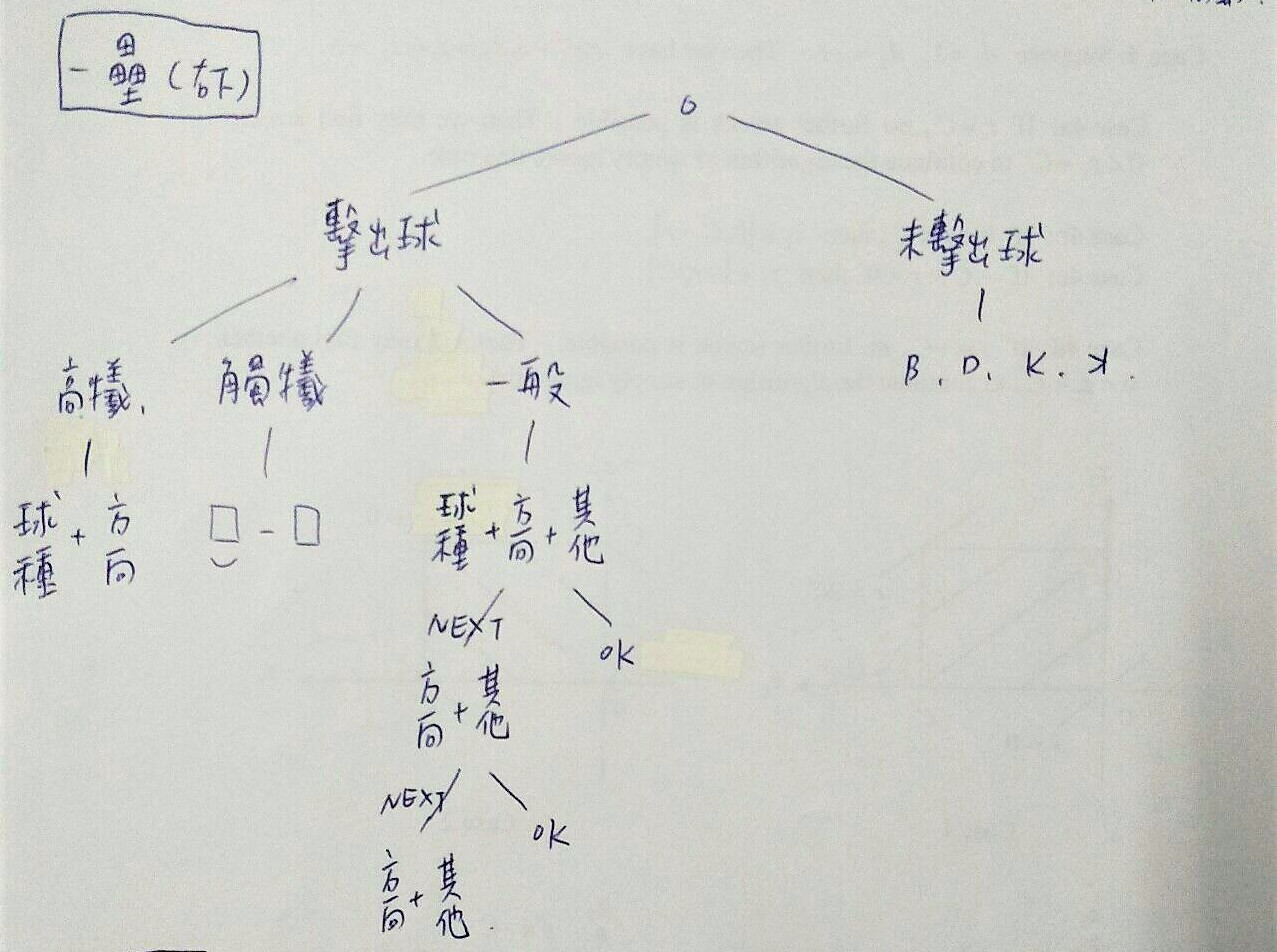
(圖四) 紀錄欄位介面設計

整理出棒球常用符號之基本功能(圖三)，每個欄位所需之符號及其記錄方式，規劃出紀錄欄位內每個欄位的介面設計(圖四)。

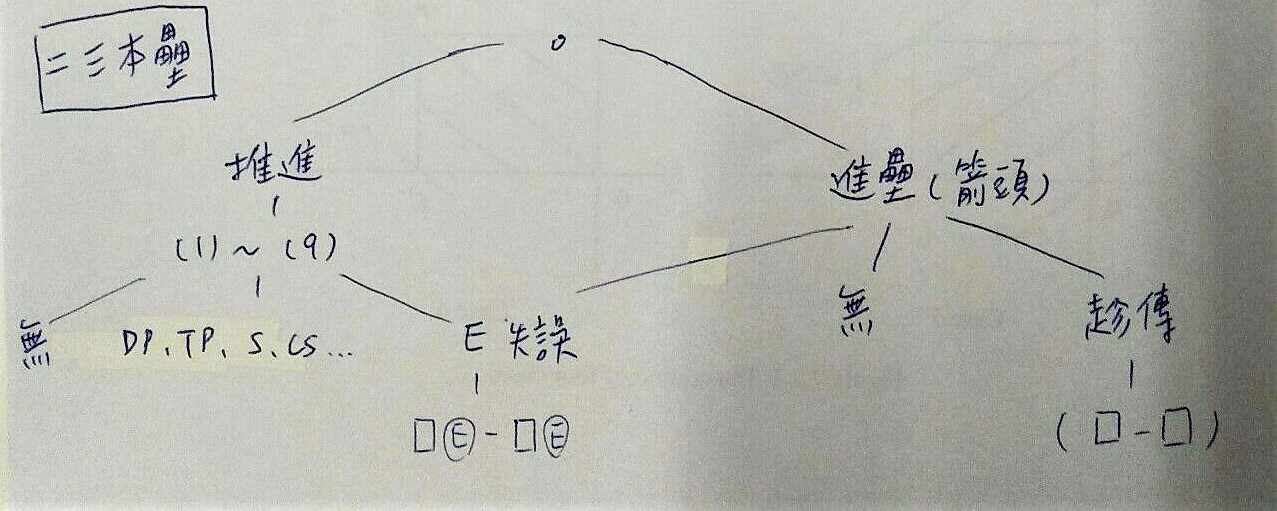
一壘為記錄打擊出去的方向、球種及上壘方式或出局方式，其他壘包則為上壘方式或出局方式，所以一壘的介面設計與其他壘包不同。



(圖五) 紀錄中間欄位決策



(圖六) 紀錄一壘欄位決策



(圖七) 紀錄二三本壘欄位決策

三、程式模組說明 (how- code)

**ScoreScrollablePanelAdapter :** 畫面元件的產生

1. 當物件顯示於畫面時被調用，可利用此方法更新該物件之內容

**public void** onBindViewHolder(RecyclerView.ViewHolder holder, **int** row, **int** column)

1. 取得item的類型

**public int** getItemViewType

1. 根據不同類型創建不同item的ViewHolder

**public** RecyclerView.ViewHolder onCreateViewHolder(ViewGroup parent, **int** viewType)

1. 設定局數顯示

**private void** setBoardNumView(**int** pos, BoardNumViewHolder viewHolder)

1. 設定球員顯示畫面，從資料庫拿球員資料，並設定資料。

**private void** setPlayerInfoView(**int** pos, **final** PlayerInfoViewHolder viewHolder)

1. 設定監聽，若點擊此區域要更改球員

viewHolder.**text\_playerName**.setOnClickListener



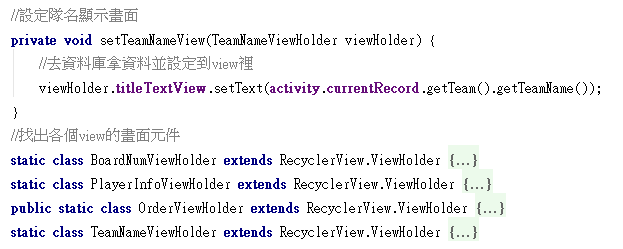
1. 設定紀錄版畫面

**private void** setOrderView(**final int** row, **final int** column, **final** OrderViewHolder viewHolder)

* 為每個區塊設定按鈕監聽



1. view的設定



**BaseCenterDialog、BaseFirstDialog、BaseOtherDialog**

點擊二三本壘出現的選單，利用RecordItem設定資料並存入資料庫以及顯示點選後的畫面。

1. 建立純文字選單，並利用switch case取得選取項目進行後續動作

builder.setItems(items, **new** DialogInterface.OnClickListener(){  
 @Override  
 **public void** onClick(DialogInterface dialog,**int** which){}}

1. 自訂dialog，根據builder設置好的一系列數據, 来建構一個dialog

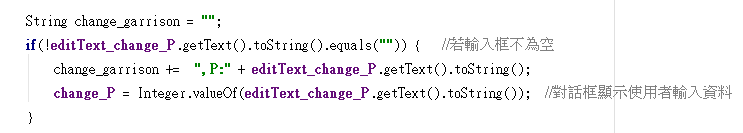
AlertDialog.Builder push\_dialog = **new** AlertDialog.Builder(**activity**);  
View view\_push\_choice1 = View.inflate(**activity**, R.layout.**record\_actionname\_dialog**, **null**);   
push\_dialog.setView(view\_push\_choice1); **final** AlertDialog new\_push\_dialog = push\_dialog.create();

1. 點取dialog內的選項後，為每一個選項設置按鈕監聽，點下後，設定recorditem的類別，並更新選取後的頁面。

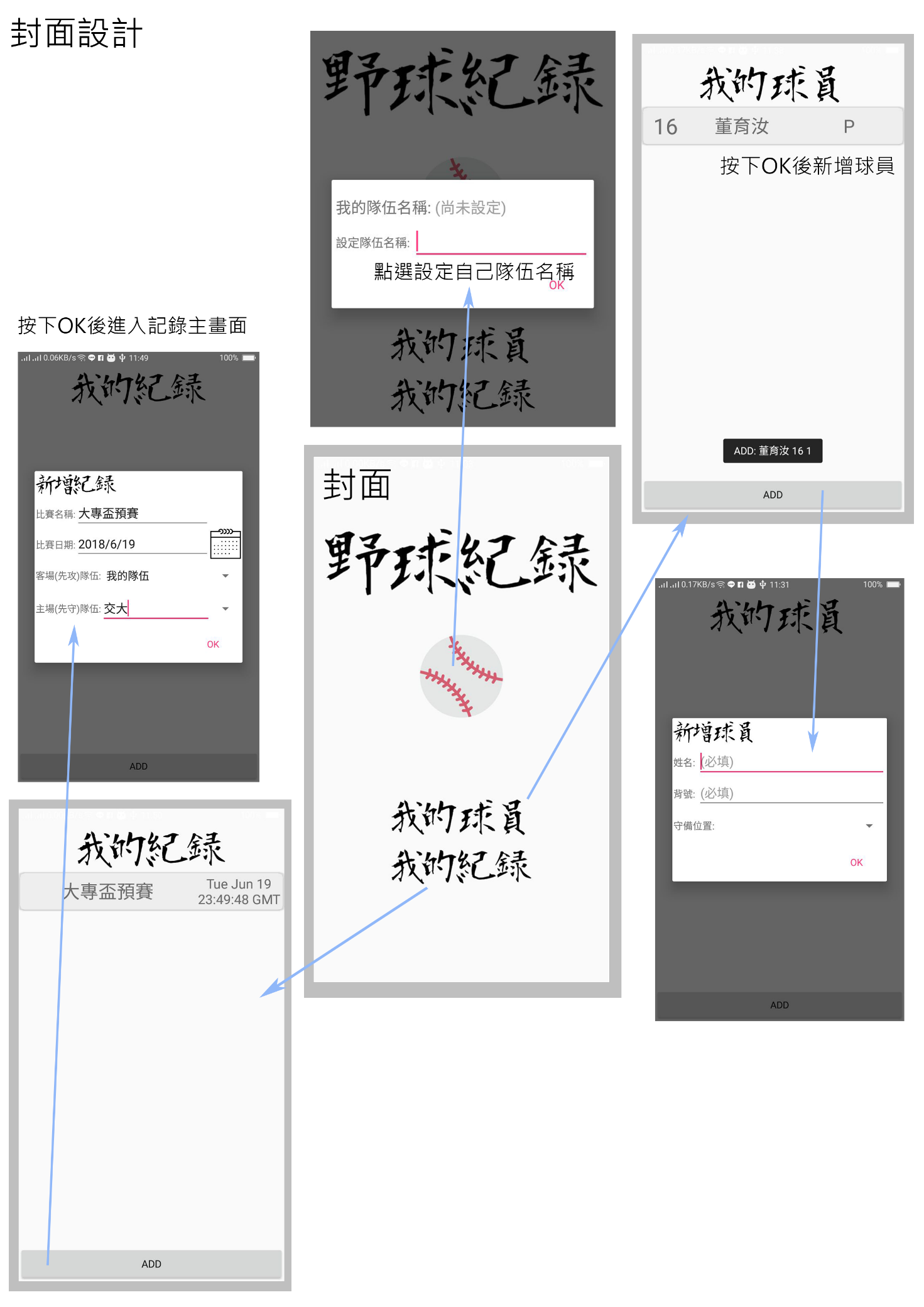
view\_push\_choice1.findViewById(R.id.***click\_dp***).setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 Toast.*makeText*(**activity**, **"雙殺DP"**, Toast.***LENGTH\_SHORT***).show();  
 new\_push\_dialog.dismiss(); *//按下選項後dialog消失* viewHolder.**recordItem**.setBallPush(RecordItem.BALL\_PUSH.***DP***); *//設定為DP類型* viewHolder.updateUI(**activity**); *//更新頁面* }  
});

1. 使用 ArrayAdapter 和字串陣列建立清單項目設定下拉式選單。
2. 點擊更換守備，將每個畫面元件設定好。

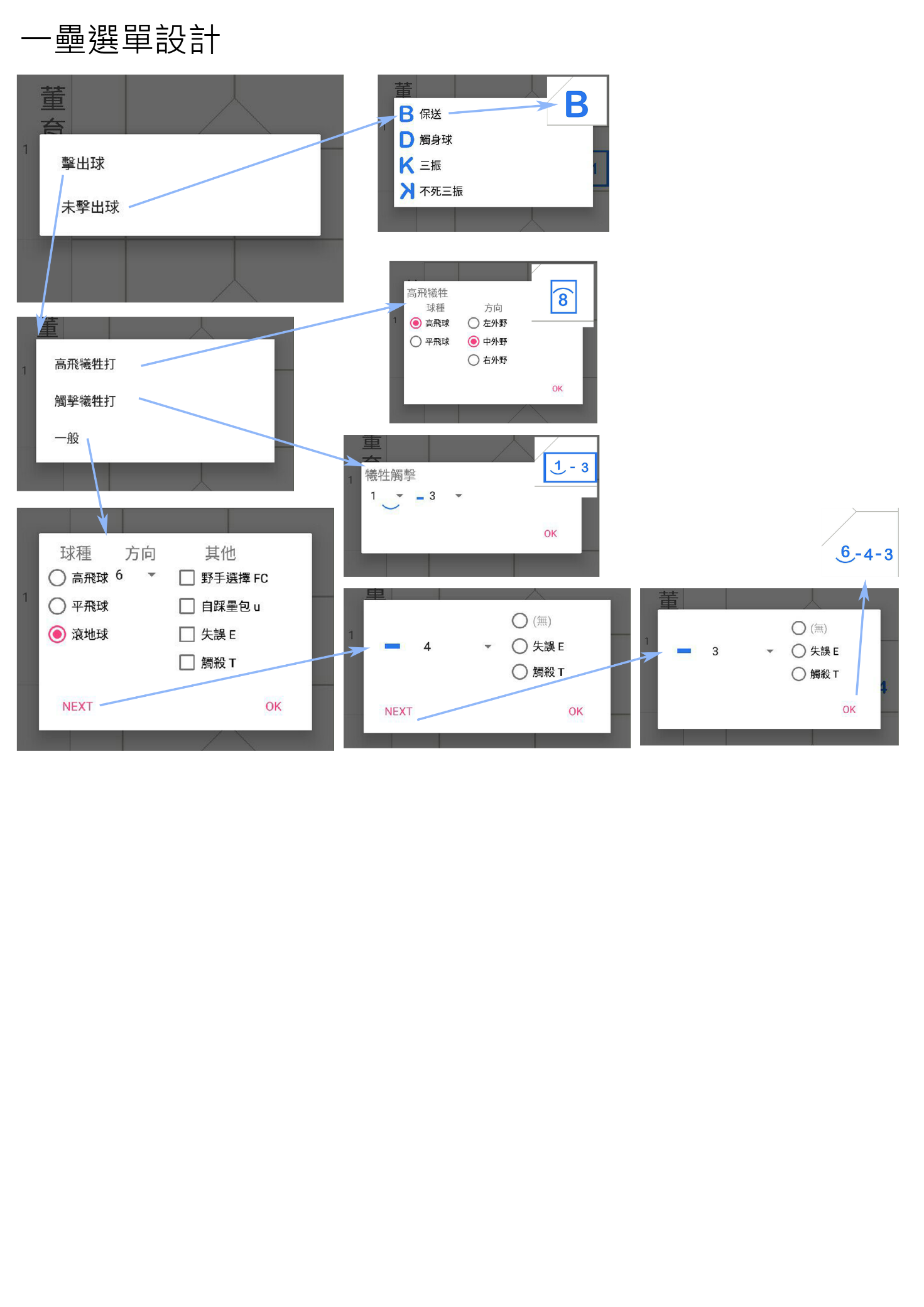
**public** voidchange\_player\_garrison(**final** ScrollablePanelAdapter.OrderViewHolder viewHolder)



四、成果 (what- result)，包含畫面的說明











五、自評

在這個野球紀錄的應用程式中，做出了大部分紙本紀錄所需要的功能，能夠透過我們做的APP顯示、儲存每筆紀錄。也改變了其他棒球紀錄應用程式的紀錄方式，不再是使用打擊者的角度紀錄，還能記錄跑壘、守備、投手狀態，讓平常就需要紙本記錄方式的人較方便、清楚，還有不做太多的限制讓使用者能自由選擇得分、安打數等，讓這種充滿變數的比賽紀錄更彈性。

美中不足的是，棒球的紀錄符號太多，比賽會出現的狀況也很多，無法在短時間就將所有的符號都加入、所有的狀況都考慮完善，所以我們選擇比賽中出現頻繁的符號來製作，但記錄整場比賽基本上是沒有問題的。