

新使用者註冊功能修正計畫

問題描述

使用者在 <http://localhost:9000/Login/New?num=1&RPC=P> 頁面 (新使用者註冊) 反映以下問題：

- 個人會員電話號碼 - 應移除必填限制 (僅限個人會員 RPC=P)
- 公司電話 - 維持必填 (公司會員 RPC=C 的「公司 (商號) 電話」仍為必填)
- 通訊地址 郵遞區號 - 目前限制 5 碼，需改為 6 碼

重要說明：本次修改僅針對個人會員 (RPC=P) 的電話號碼欄位移除必填限制，公司會員 (RPC=C) 的「公司 (商號) 電話」仍維持必填。

程式碼分析

1. 電話號碼必填限制

1.1 前端驗證 (New.cshtml)

檔案位置: [CodeBaseNew/e-service/ES/Views/Login/New.cshtml](#)

HTML 表單欄位 (第 219-225 行)

```
<label class="step-label col-sm-2" for="">
    <span style="color:red;">*</span>電話號碼</label>
    <div class="col-sm-4">
        @Html.TextBoxFor(m => m.TEL, new { @class = "form-control", maxlength = "20",
            placeholder = "請輸入您的電話號碼" })
    </div>
```

問題: 標籤前有紅色星號 * 表示必填

JavaScript 驗證函數 (第 348-354 行)

```
if (rpcType == "P") {
    // ...
    if ($("#TEL").val().trim() == "") {
        msg += "請填入電話號碼<br/>";
    }
    // ...
}
```

問題: 個人會員 (rpcType == "P") 時檢查電話號碼是否為空

1.2 後端驗證 (LoginDAO.cs)

檔案位置: CodeBaseNew/e-service/ES/DataLayers/LoginDAO.cs

ChkMemberInfo 函數 (個人會員驗證 · 第 201-204 行)

```
if (!Regex.IsMatch(detail.TEL.TONotNullString(), @"^(\d{2,4}-)(\d{6,8})?
(\#\d{1,6})?"))
{
    Msg += "電話號碼 格式錯誤 \n";
}
```

問題: 當 TEL 為空時，`TONotNullString()` 返回空字串，無法通過正則表達式驗證

1.3 Model 定義 (LoginViewModel.cs)

檔案位置: CodeBaseNew/e-service/ES/Models/LoginViewModel.cs

LoginDetailModel 類別 (第 156-161 行)

```
/// <summary>
/// 電話號碼
/// </summary>
[Display(Name = "電話號碼")]
[Required]
public string TEL { get; set; }
```

問題: 有 `[Required]` 屬性，表示必填

2. 郵遞區號 5 碼改 6 碼

2.1 前端表單欄位 (New.cshtml)

檔案位置: CodeBaseNew/e-service/ES/Views/Login/New.cshtml

個人會員地址欄位 (第 236 行)

```
@Html.TextBoxFor(m => m.ADDR_CODE, new { @id = "ADDR_CODE", @class = "form-10
form-normal", size = 10, placeholder = "郵遞區號5碼", maxlength = "5" })
```

問題: `placeholder` 顯示「郵遞區號 5 碼」，`maxlength` 限制為 5

公司會員地址欄位 (第 133 行)

```
@Html.TextBoxFor(m => m.ADDR_CODE, new { @id = "ADDR_CODE", @class = "form-10 form-normal", size = 10, placeholder = "郵遞區號5碼", maxlength = "5" })
```

問題: 同上

修正計畫

第一項：移除個人會員電話號碼必填限制

注意：本項修改僅針對個人會員 (**RPC=P**) 的電話號碼，公司會員 (**RPC=C**) 的「公司（商號）電話」維持必填不變。

步驟 1：修改前端表單 (**New.cshtml**) - 僅個人會員區塊

行號 修改內容

220 個人會員區塊：移除 `*` 紅色星號

修改前 (第 220 行 - 個人會員區塊):

```
<label class="step-label col-sm-2" for=""><span style="color:red;">*</span>電話號碼</label>
```

修改後:

```
<label class="step-label col-sm-2" for="">電話號碼</label>
```

不修改 (第 116 行 - 公司會員區塊，維持原狀):

```
<label class="step-label col-sm-2" for=""><span style="color:red;">*</span>公司（商號）電話</label>
```

步驟 2：修改前端 JavaScript 驗證 (**New.cshtml**) - 僅個人會員驗證

行號 修改內容

352-354 移除個人會員 (`rpcType == "P"`) 電話必填檢查

修改前 (第 352-354 行，在 `if (rpcType == "P")` 區塊內):

```

if ($("#TEL").val().trim() == "") {
    msg += "請填入電話號碼<br/>";
}

```

修改後:

```

// 個人會員電話號碼已改為非必填
// if ($("#TEL").val().trim() == '') {
//     msg += '請填入電話號碼<br/>';
// }

```

不修改 (第 372-374 行，在 `if (rpcType == "C")` 區塊內，維持原狀):

```

if ($("#TEL").val().trim() == "") {
    msg += "請填入公司(商號)電話號碼<br/>";
}

```

步驟 3：修改後端驗證 (LoginDAO.cs)

行號	修改內容
201-204	加入空值判斷，允許電話號碼為空

修改前:

```

if (!Regex.IsMatch(detail.TEL.TONotNullString(), @"^(\d{2,4}-)(\d{6,8})?
(\#\d{1,6})?#$"))
{
    Msg += "電話號碼 格式錯誤 \n";
}

```

修改後:

```

// 電話號碼改為非必填，僅在有填寫時驗證格式
if (!string.IsNullOrEmpty(detail.TEL))
{
    if (!Regex.IsMatch(detail.TEL.TONotNullString(), @"^(\d{2,4}-)(\d{6,8})?
(\#\d{1,6})?#$"))
    {
        Msg += "電話號碼 格式錯誤 \n";
    }
}

```

步驟 4：修改 Model (LoginViewModel.cs)

行號 修改內容

160 移除 [Required] 屬性

修改前:

```
[Display(Name = "電話號碼")]
[Required]
public string TEL { get; set; }
```

修改後:

```
[Display(Name = "電話號碼")]
// [Required] // 電話號碼改為非必填
public string TEL { get; set; }
```

第二項：郵遞區號改為 6 碼

步驟 1：修改前端表單 (New.cshtml)

行號 修改內容

133 公司會員：placeholder 改為「郵遞區號 6 碼」，maxlength 改為 "6"

236 個人會員：placeholder 改為「郵遞區號 6 碼」，maxlength 改為 "6"

修改前 (第 133 行 - 公司會員):

```
@Html.TextBoxFor(m => m.ADDR_CODE, new { @id = "ADDR_CODE", @class = "form-10
form-normal", size = 10, placeholder = "郵遞區號5碼", maxlength = "5" })
```

修改後:

```
@Html.TextBoxFor(m => m.ADDR_CODE, new { @id = "ADDR_CODE", @class = "form-10
form-normal", size = 10, placeholder = "郵遞區號6碼", maxlength = "6" })
```

修改前 (第 236 行 - 個人會員):

```
@Html.TextBoxFor(m => m.ADDR_CODE, new { @id = "ADDR_CODE", @class = "form-10
form-normal", size = 10, placeholder = "郵遞區號5碼", maxlength = "5" })
```

修改後:

```
@Html.TextBoxFor(m => m.ADDR_CODE, new { @id = "ADDR_CODE", @class = "form-10 form-normal", size = 10, placeholder = "郵遞區號6碼", maxlength = "6" })
```

完整修改檔案清單

經過深入分析程式碼，以下是所有需要修改的檔案完整清單：

一、個人會員電話號碼移除必填限制

重要：僅修改個人會員 (RPC=P) 的電話號碼，公司會員 (RPC=C) 的「公司（商號）電話」維持必填。

1. 前端視圖檔案

檔案路徑	行號	修改內容
ES/Views/Login/New.cshtml	220	移除紅色星號 * (個人會員電話號碼欄位)
ES/Views/Login/New.cshtml	352-354	移除 JavaScript 必填驗證 (個人會員區塊)
ES/Views/Login/Edit1.cshtml	255	移除紅色星號 * (個人會員電話號碼欄位)
ES/Views/Login/Edit1.cshtml	431-433	移除 JavaScript 必填驗證 (個人會員區塊)

不修改的檔案 (公司會員電話維持必填):

檔案路徑	行號	說明
ES/Views/Login/New.cshtml	116	公司（商號）電話標籤的紅色星號 * 維持不變
ES/Views/Login/New.cshtml	372-374	公司會員 JavaScript 必填驗證維持不變

Edit1.cshtml 第 255 行 修改前:

```
<label class="step-label col-sm-2" for=""><span style="color:red;">*</span>電話號碼</label>
```

修改後:

```
<label class="step-label col-sm-2" for="">電話號碼</label>
```

Edit1.cshtml 第 431-433 行 修改前:

```

if ($("#TEL").val().trim() == "") {
    msg += "請填入電話號碼<br/>";
}

```

修改後:

```

// 電話號碼已改為非必填
// if ($("#TEL").val().trim() == '') {
//     msg += '請填入電話號碼<br/>';
// }

```

2. 後端資料存取層

檔案路徑	行號	修改內容
ES/DataLayers/LoginDAO.cs	201-204	ChkMemberInfo() 函數：加入空值判斷
ES/DataLayers/LoginDAO.cs	932-935	另一處個人帳號 TEL 驗證：移除或改為非必填

LoginDAO.cs 第 932-935 行 修改前:

```

if (string.IsNullOrEmpty(member.TEL))
{
    msg += "請輸入電話號碼\n";
}

```

修改後:

```

// 電話號碼改為非必填
// if (string.IsNullOrEmpty(member.TEL))
// {
//     msg += "請輸入電話號碼\n";
// }

```

3. Model 定義

需要修改的 **Model** (僅個人會員相關):

檔案路徑	行號	類別名稱	修改內容
ES/Models/LoginViewModel.cs	160	LoginDetailModel	移除 [Required] 屬性

檔案路徑	行號	類別名稱	修改內容
ES/Models/MemberModels.cs	115	PersonalMember	移除 [Required] 屬性 (個人會員)
ES/Models/MemberModels.cs	306	PersonalMemberForApp	移除 [Required] 屬性 (個人會員 App)

不修改的 Model (公司會員電話維持必填):

檔案路徑	行號	類別名稱	說明
ES/Models/MemberModels.cs	182	CompanyMember	維持 [Required] 屬性不變
ES/Models/MemberModels.cs	365	CompanyMemberForApp	維持 [Required] 屬性不變

MemberModels.cs 第 115 行 (PersonalMember 類別) 修改前:

```
[Required(ErrorMessage = "請輸入電話號碼")]
[RegularExpression(@"([0][0-9]{1,3}[-])?[0-9]{5,8}([#][0-9]{1,5})?", ErrorMessage =
= "電話號碼格式錯誤")]
[Display(Name = "電話號碼")]
public override string Tel { get; set; }
```

修改後:

```
// [Required(ErrorMessage = "請輸入電話號碼")] // 個人會員電話號碼改為非必填
[RegularExpression(@"([0][0-9]{1,3}[-])?[0-9]{5,8}([#][0-9]{1,5})?", ErrorMessage =
= "電話號碼格式錯誤")]
[Display(Name = "電話號碼")]
public override string Tel { get; set; }
```

MemberModels.cs 第 182 行 (CompanyMember 類別) - 不修改，維持原狀:

```
[Required(ErrorMessage = "請輸入公司 (商號) 電話")]
[RegularExpression(@"([0][0-9]{1,3}[-])?[0-9]{5,8}([#][0-9]{1,5})?", ErrorMessage =
= "電話格式錯誤")]
[Display(Name = "公司 (商號) 電話")]
public override string Tel { get; set; }
```

MemberModels.cs 第 306 行 (PersonalMemberForApp 類別) 修改前:

```
[Required(ErrorMessage = "請輸入電話號碼")]
[RegularExpression(@"([0][0-9]{1,3}[-])?[0-9]{5,8}([#][0-9]{1,5})?", ErrorMessage =
= "電話號碼格式錯誤")]
```

```
[Display(Name = "電話號碼")]
public override string Tel { get; set; }
```

修改後：

```
// [Required(ErrorMessage = "請輸入電話號碼")] // 個人會員電話號碼改為非必填
[RegularExpression(@"([0][0-9]{1,3}[-])?[0-9]{5,8}([#][0-9]{1,5})?", ErrorMessage =
= "電話號碼格式錯誤")]
[Display(Name = "電話號碼")]
public override string Tel { get; set; }
```

MemberModels.cs 第 365 行 (CompanyMemberForApp 類別) - 不修改，維持原狀：

```
[Required(ErrorMessage = "請輸入公司 (商號) 電話")] // 公司電話維持必填
[RegularExpression(@"([0][0-9]{1,3}[-])?[0-9]{5,8}([#][0-9]{1,5})?", ErrorMessage =
= "電話格式錯誤")]
[Display(Name = "公司 (商號) 電話")]
public override string Tel { get; set; }
```

二、郵遞區號改為 6 碼

1. 前端視圖檔案

檔案路徑	行號	修改內容
ES/Views/Login/New.cshtml	133	公司會員 : placeholder="郵遞區號6碼", maxlength="6"
ES/Views/Login/New.cshtml	236	個人會員 : placeholder="郵遞區號6碼", maxlength="6"
ES/Views/Login/Edit1.cshtml	103	公司會員 : placeholder="郵遞區號6碼", maxlength="6"
ES/Views/Login/Edit1.cshtml	271	個人會員 : placeholder="郵遞區號6碼", maxlength="6"

Edit1.cshtml 第 103 行 修改前：

```
@Html.TextBoxFor(m => m.ADDR_CODE, new { @class = "form-10 form-normal", size =
10, placeholder = "郵遞區號5碼", maxlength = "5" })
```

修改後：

```
@Html.TextBoxFor(m => m.ADDR_CODE, new { @class = "form-10 form-normal", size =
10, placeholder = "郵遞區號6碼", maxlength = "6" })
```

Edit1.cshtml 第 271 行 修改前:

```
@Html.TextBoxFor(m => m.ADDR_CODE, new { @id = "ADDR_CODE", @class = "form-10 form-normal", size = 10, placeholder = "郵遞區號5碼", maxlength = "5" })
```

修改後:

```
@Html.TextBoxFor(m => m.ADDR_CODE, new { @id = "ADDR_CODE", @class = "form-10 form-normal", size = 10, placeholder = "郵遞區號6碼", maxlength = "6" })
```

2. Helper 擴充方法 (可選)

檔案路徑	行號	修改內容
ES/Helpers/AddrExtension.cs	67	maxlength = 10 (目前已是 10 · 無需修改)
ES/Helpers/AddrExtension.cs	200	maxlength = 10 (目前已是 10 · 無需修改)

備註: AddrExtension.cs 中的 maxlength 已設定為 10 · 足以容納 6 碼郵遞區號 · 無需修改。

修改檔案總覽

序號	檔案路徑	修改項目
1	ES/Views/Login/New.cshtml	個人會員電話號碼標籤、JS 驗證 (個人會員區塊) 、郵遞區號欄位
2	ES/Views/Login/Edit1.cshtml	個人會員電話號碼標籤、JS 驗證 (個人會員區塊) 、郵遞區號欄位
3	ES/DataLayers/LoginDAO.cs	ChkMemberInfo() 個人會員電話驗證 (第 201-204 行) 、個人帳號 TEL 驗證 (第 932-935 行)
4	ES/Models/LoginViewModel.cs	TEL 屬性的 [Required] 註解 (第 160 行)
5	ES/Models/MemberModels.cs	僅個人會員 Tel 屬性的 [Required] 註解 (第 115、306 行) · 公司會員維持不變 (第 182、365 行)

修改影響範圍

功能頁面

- 新使用者註冊頁面 ([/Login/New](#))
- 會員資料編輯頁面 ([/Login/Edit1](#))

會員類型

- 個人會員 (RPC=P) - 電話號碼改為非必填
 - 公司會員 (RPC=C) - 公司 (商號) 電話維持必填
-

測試建議

1. 個人會員電話號碼測試 (RPC=P):

- 測試不填寫電話號碼是否可以成功註冊
- 測試填寫正確格式電話號碼是否可以成功註冊
- 測試填寫錯誤格式電話號碼是否會顯示錯誤訊息

2. 公司會員電話號碼測試 (RPC=C):

- 測試不填寫公司 (商號) 電話應該顯示必填錯誤訊息
- 測試填寫正確格式電話號碼是否可以成功註冊
- 測試填寫錯誤格式電話號碼是否會顯示錯誤訊息

3. 郵遞區號測試:

- 測試填寫 6 碼郵遞區號是否可以正常輸入
 - 測試郵遞區號查詢功能是否正常運作
 - 測試是否能成功儲存 6 碼郵遞區號
-

附錄：郵遞區號 6 碼資料庫分析

資料表結構 (已驗證)

郵遞區號存儲在 **ZIPCODE** 資料表中，經實際查詢 SQL Server 資料庫確認欄位結構如下：

欄位名稱	資料型態	最大長度	說明
ZIP_CO	nvarchar	10	郵遞區號 (主鍵)
CITYNM	nvarchar	20	縣市名稱
TOWNNM	nvarchar	20	鄉鎮市區名稱
ROADNM	nvarchar	50	街道名稱
ROUND	nvarchar	50	範圍說明

驗證結果: ZIP_CO 欄位為 nvarchar(10)，足夠存放 6 碼郵遞區號，無需修改資料庫結構。

Entity 類別

檔案位置: e-service/ES/Models/Entities/ZIPCODE.cs

```
public class TblZIPCODE
{
    public string ZIP_CO { get; set; } // 郵遞區號
    public string CITYNM { get; set; } // 縣市
```

```
public string TOWNNM { get; set; } // 鄉鎮  
public string ROADNM { get; set; } // 街道  
public string ROUND { get; set; } // 範圍  
}
```

使用方式

系統透過 **ZIP_CO** 欄位查詢郵遞區號對應的縣市、鄉鎮資訊：

```
TblZIPCODE zip = new TblZIPCODE();  
zip.ZIP_CO = model.ADDR_CODE; // 設定郵遞區號  
var zipdata = dao.GetRow(zip); // 查詢對應資料  
string address = zipdata.CITYNM + zipdata.TOWNNM; // 組合地址
```

資料庫相容性評估

- **ZIP_CO** 欄位為 **nvarchar(10)** 型態，可容納 6 碼郵遞區號
- 現有 SQL 查詢使用 **SUBSTRING(ZIP_CO, 1, 3)** 取前 3 碼作為鄉鎮代碼
- 6 碼郵遞區號格式相容現有查詢邏輯（前 3 碼仍為鄉鎮代碼）

結論

1. **資料庫欄位長度足夠**：**ZIP_CO** 欄位為 **nvarchar(10)**，可存放 6 碼郵遞區號
2. 如需完整支援 6 碼郵遞區號，需匯入最新的 6 碼郵遞區號對照表資料

文件建立日期: 2025-12-03 文件更新日期: 2025-12-04