|  |  |  |  |
| --- | --- | --- | --- |
| **Subject :** | QPlayPESPIS QAccountManage | **Latest Ver :** | 1.9.1 |
| **Latest Release Date :** | 2019-02-13 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Change History** | | | | |
| Date | Author | Version | Type | Comment |
| 2019-02-13 | Darren.K.Ti | 1.9.1 | Function  DB | 1. QAccountManage PESPIS Ver 1.9.1 |

# 目錄

[**目錄**](#_30j0zll) **2**

[**1.0 系統設計 (System Design)**](#_3znysh7) **3**

[**2.0 資料庫設計 (Database Design)**](#_tyjcwt) **3**

[**3.0 Implementation**](#_3dy6vkm) **4**

[**4.0 功能修改記錄**](#_1t3h5sf) **5**

[[ QAccount Management ]](#_4d34og8) 5

[4.1 需求說明](#_sh3w8xzthq6) 5

[4.2 時程評估 (依照開發順序)](#_u8h9nl3u7gm5) 5

[4.3 對應文件](#_n13lnemcda1u) 5

[4.4 修改項目](#_yppwycepxn3w) 5

[4.5 功能架構圖](#_fftjyhwykpuh) 6

[4.6 資料庫物件清單](#_lzmsqs8dyqwn) 6

[4.7 實體關聯圖](#_k9qelncd1e7h) 6

[4.8 資料表結構](#_egrqds71as4x) 6

[4.9 公用元件](#_q7h41ae7amg6) 6

[4.10 共用介面](#_9lwuatinxq6n) 6

[4.11 資料流程圖 Data Flow Diagram](#_bshrfzvgoch1) 6

[4.12 程式碼修改記錄](#_tbufotq7rlf3) 8

[4.13 UI 示意圖](#_tdi1di175fja) 9

[4.14 Test Case](#_iluhyr3tobf) 9

# 1.0 系統設計 (System Design)

**1.1 系統架構圖 (System Framework Diagram)**

* QPlay Backend Platform 整體的系統架構圖，待補上

**1.2 系統流程圖 (System Flow)**

* QChat API 基於 laravel 5.2 框架進行開發，並使用 Repository + Service 模式之程式架構。
* 使用 index.php 作為進入點，route 為路由分配，midware 作為中介層，進行 request 和 response 的驗證，以及寫入 API Log。



**1.3 類別圖 (Package / Class Diagram)**

* QPlay Backend Platform 整體的類別圖，待補上

# 2.0 資料庫設計 (Database Design)

請參考資料庫設計文件 : QPlay\_DB\_Schema.docx

2.1 資料庫物件清單 (Database Object List)

* 每個修改項目 4.X 各自描述

2.2 實體關聯圖 (Entity-Relation Chart)

* 每個修改項目 4.X 各自描述

2.3 資料表結構 (Table Layout)

* 每個修改項目 4.X 各自描述

# 3.0 Implementation

3.1 公用元件 (Public Component)

* 每個修改項目 4.X 各自描述

3.2 共用介面 (Public Interface)

* 每個修改項目 4.X 各自描述

# 4.0 功能修改記錄

## [ QAccount Management ]

### 4.1 需求說明

* **QPlay 後台新增功能**
  + **用戶管理 - QAccount 管理**
    - **手動新增 QAccount**
      * **必填資料**
        + **loign\_id : 不能與現有資料重複**
        + **emp\_no : 不能與現有資料重複**
        + **emp\_name**
        + **user\_domain : 使用下拉選單，選擇目前所有 domain**
        + **company : 使用下拉選單，選擇目前所有 company**
      * **選填資料**
        + **ext\_no**
        + **email**
      * **Hardcode 資料**
        + **source\_from = qplay**
    - **只管理透過 [ QAccount 管理 ] 手動新增的資料**
      * **排除 [ User Sync Job ] 和 [ QPay Shop ]**
    - **[ QAccount 管理 ] 新增的帳號，無法使用 QPay**
  + **初期，只用來維護 (茶水總機) 的帳號，之後在應用到其他地方**
* 備註
  + QAccount 來源
    - User Sync Job
    - QPay Shop
    - QAccount Management

### 4.2 時程評估 (依照開發順序)

* [ **Routes 整理 / 部分舊有 controller 更新至新增的 accountController** ]
  + 預估時程 : ? ~ ? (1 day)
  + 實際時程 :
  + 備註 :
    - 更新 controller 之後，測試功能是否正常
* [ **新增頁面 (QAccount 管理) - View QAccountList / QPlay 後台參數設定** ]
  + 預估時程 : ? ~ ? (1.5 day)
  + 實際時程 :
  + 備註 :
    - 包含
      * 開發 **qaccount\_list.blade.php** 的 HTML / CSS / JS
        + 參考現有 qpay\_maintain/ qpay\_user\_maintain/ **shop.blade.php**
      * 開發下列 Function
        + getCompanyList
        + getDomainList
      * QPlay 後台的選單設定和群組維護
* [ **功能開發 - getQAccountList / Error Code 和 語言檔案更新** ]
  + 預估時程 : ? ~ ? (1 day)
  + 實際時程 :
  + 備註 :
    - 包含
      * 前端 JS 處理
      * 後端 PHP 處理
      * Error Code 和 語言檔案更新
* [ **功能開發 - newQAccount** ]
  + 預估時程 : ? ~ ? (1 day)
  + 實際時程 :
  + 備註 :
    - 包含
      * 前端 JS 處理
      * 後端 PHP 處理
* [ **功能開發 - updateQAccount** ]
  + 預估時程 : ? ~ ? (1 day)
  + 實際時程 :
  + 備註 :
    - 包含
      * 前端 JS 處理
      * 後端 PHP 處理

### 4.3 對應文件

* 無

### 4.4 修改項目

* qplay/
  + 修改
    - app/ Http/ **routes.php**
    - app/ Http/ Controllers/
      1. **platformController.php**
      2. **qpayController.php**
    - app/ Services/ **QPayShopService.php**
    - app/ Repositories/ **UserRepository.php**
    - app/ lib/ **ResultCode.php**
    - resources/ lang
      1. en/ **messages.php**
      2. zh-cn/ **messages.php**
      3. zh-tw/ **messages.php**
    - resources/ views/ qpay\_maintain/ qpay\_user\_maintain/
      1. **shop.blade.php**
  + 新增
    - app/ Http/ Controllers/ **accountController.php**
    - app/ Services/ **UserService.php**
    - resources/ views/ user\_maintain/
      1. **qaccount\_list.blade.php**

### 4.5 功能架構圖

* 無

### 4.6 資料庫物件清單

|  |  |  |  |
| --- | --- | --- | --- |
| **DB Type** | **Type** | **Object Name** | **Description** |
| MySQL | InnoDB | qp\_user | 員工基本資料 |
| MySQL | InnoDB | qp\_password\_log | [QAccount 登入密碼] 的修改記錄 |

### 4.7 實體關聯圖

* 無

### 4.8 資料表結構

* 參考文件 :
  + 無

### 4.9 公用元件

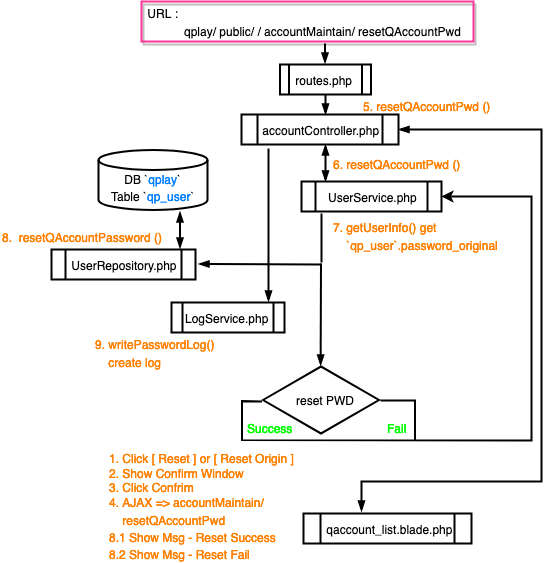
* 內部模組

### 4.10 共用介面

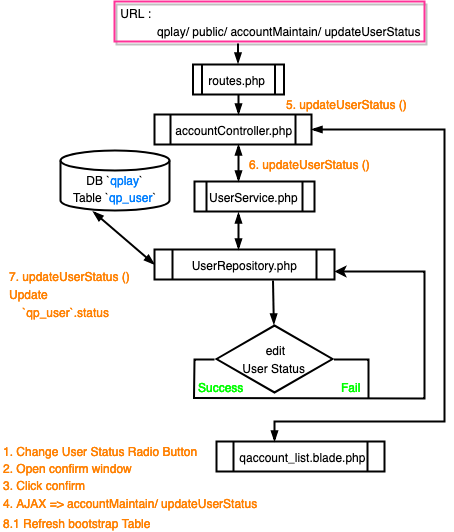
* 無

### 4.11 資料流程圖 Data Flow Diagram

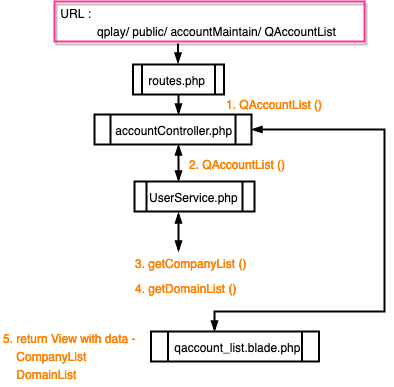
#### Routes : resetQAccountPwd



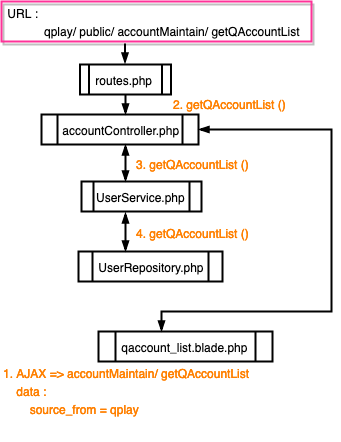
#### Routes : updateUserStatus



#### Routes : QAccountList



#### Routes : getQAccountList



#### Routes : newQAccount



#### Routes : updateQAccount



### 4.12 程式碼修改記錄

##### app/ Http/ **routes.php**

##### URL Request Handle 修改下列 Route :

##### [ 用戶管理 ] 相關的 Route，原先使用 platformController.php，改為使用新增的 accountController.php

##### [ 消費券管理 - 用戶管理 - 店家資料 ] 的

##### resetQAccountPwd，原先使用 qpayController.php，改為使用新增的 accountController.php

##### updateUserStatus，原先使用 qpayController.php，改為使用新增的 accountController.php

##### 使用 accountController.php

##### prefix : accountMaintain/

##### getUserList，取得 User 資料列表，加上條件式。

##### getUserListWithoutGroup，取得 User 資料列表。

##### getRoleList，取得 企業角色 資料列表，加上條件式。

##### removeUserRight，移除 User 的 QPlay 使用權限。

##### saveUser，儲存 User 資料。

##### deleteRole，移除 企業角色 資料。

##### saveRole，儲存 企業角色 資料。

##### getRoleUsers，取得 企業角色 底下的所有 User 資料列表。

##### saveRoleUsers，儲存 企業角色 底下的所有 User 設定。

##### resetQAccountPwd，重置 or 更新 QAccount 密碼。

##### updateUserStatus，更新 User 的 QPlay 使用狀態。

##### URL Request Handle 新增下列 Route :

##### 使用 accountController.php

##### prefix : accountMaintain/

##### QAccountList，Return View ‘user\_maintain/ qaccount\_list’。

##### getQAccountList，取得 QAccount 資料列表。

##### newQAccount，新增 QAccount 資料。

##### updateQAccount，更新 QAccount 資料。

* app/ Http/ Controllers/

##### **platformController.php**

* + - 下列現有 function，改為寫在 accountController.php
      * getUserList ()
      * getUserListWithoutGroup ()
      * getRoleList ()
      * removeUserRight ()
      * saveUser ()
      * deleteRole ()
      * saveRole ()
      * getRoleUsers ()
      * saveRoleUsers ()

##### **qpayController.php** 現有 function

* + - resetQAccountPwd ()
      * 移除此 function 改為寫在 accountController.php
    - updateUserStatus ()
      * 移除此 function 改為寫在 accountController.php

##### **accountController.php** 定義 function

* + - QAccountList ()
      * 使用 UserService.php 定義的 getCompanyList () 取得 Company List。
      * 使用 UserService.php 定義的 getDomainList () 取得 Domain List。
      * Return View ( ‘user\_maintain/ qaccount\_list’ ) With Data
        + companyList
        + domainList
    - getQAccountList ()
      * 帶入
        + source\_from : required
      * 使用 UserService.php 定義的 getQAccountList ()
    - newQAccount ()
      * 帶入
        + login\_id : required
        + emp\_no : required
        + emp\_name : required
        + qaccount\_pwd : required
        + user\_domain : required
        + company : required
        + email : optional
        + ext\_no : optional
      * 使用 UserService.php 定義的 newQAccount ()
    - updateQAccount ()
      * 帶入
        + login\_id : required
        + emp\_no : required
        + emp\_name : required
        + user\_domain : required
        + company : required
        + email : optional
        + ext\_no : optional
      * 使用 UserService.php 定義的 updateQAccount ()
    - resetQAccountPwd ()
      * 從 qpayController.php 複製此 function 的內容，並且修改
        + 原先

使用 qpayShopService->resetQAccountPwd ()

* + - * + 改為

使用 userService->resetQAccountPwd ()

* + - updateUserStatus ()
      * 從 qpayController.php 複製此 function 的內容，並且修改
        + 原先

使用 qpayShopService->updateUserStatus ()

* + - * + 改為

使用 userService->updateUserStatus ()

* app/ Services/

##### **QPayShopService.php** 現有 Function

* + - resetQAccountPwd ()
      * 移除此 function 改為寫在 UserService.php
    - updateUserStatus ()
      * 移除此 function 改為寫在 UserService.php

##### **UserService.php** 定義 Function

* + - getCompanyList ()
      * 使用 UserRepository.php定義的 getCompanyList ()。
      * 回傳
        + `qp\_user`.company 資料列表
    - getDomainList ()
      * 使用 UserRepository.php定義的 getDomainList ()。
      * 回傳
        + `qp\_user`.user\_domain 資料列表
    - getQAccountList ()
      * 使用 UserRepository.php定義的 getQAccountList ()。
      * 帶入
        + source\_from
      * 回傳
        + `qp\_user` 資料列表
    - newQAccount ()
      * 帶入
        + login\_id
        + emp\_no
        + emp\_name
        + qaccount\_pwd
        + user\_domain
        + company
        + email
        + ext\_no
      * 使用 UserRepository.php定義的 countUserData () 檢查
        + login\_id 是否已經存在

否

繼續往下執行

是

回傳 000925\_userLoginIDExist

* + - * + emp\_no 是否已經存在

否

繼續往下執行

是

回傳 000926\_userEmpNoExist

* + - * 使用 UserRepository.php定義的 newUser ()。
        + 帶入下列參數

login\_id

emp\_no

emp\_name

user\_domain

company

email

ext\_no

password = password\_hash( qaccount\_pwd )

password\_original = password\_hash( qaccount\_pwd )

trade\_pwd\_original = NULL

site\_code = NULL

department = NULL

status = Y

resign = N

ad\_flag = N

source\_from = qplay

created\_at = date( ‘Y-m-d H:i:s’, time() )

created\_user = Auth::user()->row\_id

* + - * 回傳
        + 處理結果
    - updateQAccount ()
      * 帶入
        + login\_id
        + emp\_no
        + emp\_name
        + user\_domain
        + company
        + email
        + ext\_no
      * 使用 UserRepository.php定義的 countUserData () 檢查
        + login\_id 是否已經存在

否

繼續往下執行

是

回傳 000925\_userLoginIDExist

* + - * + emp\_no 是否已經存在

否

繼續往下執行

是

回傳 000926\_userEmpNoExist

* + - * 使用 UserRepository.php定義的 updateUser ()。
        + 帶入下列參數

login\_id

emp\_no

emp\_name

user\_domain

company

email

ext\_no

updated\_at = date( ‘Y-m-d H:i:s’, time() )

* + - * 回傳
        + 處理結果
    - resetQAccountPwd ()
      * 從 qpayShopService.php 複製此 function 的內容
    - updateUserStatus ()
      * 從 qpayShopService.php 複製此 function 的內容
* app/ Repositories/

##### **UserRepository.php** 定義 Function

* + - getCompanyList ()
      * 取得 `qp\_user` 中所有的 company 資料 (排除 cm , shop, HR)
      * SQL
        + SELECT DISTINCT(company) FROM `qp\_user` WHERE company NOT IN (cm, shop, HR)
      * 回傳
        + 資料列表
    - getDomainList ()
      * 取得 `qp\_user` 中所有的 user\_domain 資料 (排除 cm , shop)
      * SQL
        + SELECT DISTINCT(user\_domain) FROM `qp\_user` WHERE user\_domain NOT IN (cm, shop)
      * 回傳
        + 資料列表
    - getQAccountList ()
      * 取得指定 source from 的 QAccount 資料
      * 帶入
        + source\_from
      * 條件
        + `qp\_user`.source\_from = source\_from
      * 回傳
        + 資料列表
    - countUserData ()
      * 取得 `qp\_user` 指定欄位的資料筆數
      * 帶入
        + column name
        + value
      * 條件
        + `qp\_user`.column\_name = value
      * 回傳
        + 資料筆數
* app/ lib/ **ResultCode.php**
  + 新增
    - 000925\_userLoginIDExist // 員工 Login ID 已經存在
    - 000926\_userEmpNoExist // 員工 Emp No 已經存在
* resources/ lang 新增資料

##### en/ **messages.php**

* + - USER\_LOGIN\_ID\_EXIST => User Login ID has exist.
    - USER\_EMP\_NO\_EXIST => User Emp No. has exist.
    - USER\_EXT\_NO => Ext No.
    - TITLE\_USER\_QACCOUNT\_MAINTAIN => QAccount Maintain
    - UPDATE\_USER\_DATA => Update User Data.

##### zh-cn/ **messages.php**

* + - USER\_LOGIN\_ID\_EXIST => 员工帐号已经存在
    - USER\_EMP\_NO\_EXIST => 员工编号已经存在
    - USER\_EXT\_NO => 分机电话
    - TITLE\_USER\_QACCOUNT\_MAINTAIN => QAccount 管理
    - UPDATE\_USER\_DATA => 修改员工资料

##### zh-tw/ **messages.php**

* + - USER\_LOGIN\_ID\_EXIST => 員工帳號已經存在
    - USER\_EMP\_NO\_EXIST => 員工編號已經存在
    - USER\_EXT\_NO => 分機電話
    - TITLE\_USER\_QACCOUNT\_MAINTAIN => QAccount 管理
    - UPDATE\_USER\_DATA => 修改員工資料
* resources/ views/ qpay\_maintain/ qpay\_user\_maintain

##### **shop.blade.php**

* + - 修改現有程式碼
    - 下列 2 個 AJAX 的 url 加上 prefix
      * resetQAccountPwd => accountMaintain/resetQAccountPwd
      * updateUserStatus => accountMaintain/updateUserStatus
* resources/ views/ user\_maintain/

##### **qaccount\_list.blade.php**

* + - 比照 [ 消費券管理 - 用戶管理 - 店家資料 ] 的程式碼以及 UI
      * 參考 resources/ views/ qpay\_maintain/ qpay\_user\_maintain/ **shop.blade.php**
    - 新增資料的 UI
      * 新增欄位
        + user domain : 使用 UserService.php 定義的 getDomainList () 回傳 Domain List
        + company : 使用 UserService.php 定義的 getCompanyList () 回傳 Company List
        + email : 選填
      * 移除欄位
        + 地址 : 不需要此欄位
        + QPay 交易密碼 : 不需要此欄位
      * 資料處理
        + AJAX => URL : accountMaintain/ newQAccount 新增資料
    - 編輯資料的 Dialog UI
      * 新增欄位
        + emp\_no
        + user domain : 使用 UserService.php 定義的 getDomainList () 回傳 Domain List
        + company : 使用 UserService.php 定義的 getCompanyList () 回傳 Company List
        + email
      * 移除欄位
        + 地址 : 不需要此欄位
      * 資料處理
        + AJAX => URL : accountMaintain/ updateQAccount 更新資料
    - 資料列表的 UI
      * 新增欄位
        + user domain
        + company
      * 移除欄位
        + 地址 : 不需要此欄位
        + 電話 : 不需要此欄位
        + [ 更新交易密碼 ] : 不需要此功能
        + [ 交易狀態 ] 的設定 : 不需要此功能
      * 欄位順序
        + emp\_name
        + login\_id
        + emp\_no
        + user\_domain
        + company
        + [ 更新登入密碼 ]
        + [ 重置登入密碼 ]
        + [ 使用狀態 ]
      * 資料處理
        + AJAX => URL : accountMaintain/ getQAccountList 取得資料

帶入參數

source\_from = qplay (只管理 qplay 來源的資料)

* + - * + AJAX => URL : accountMaintain/ updateQAccount 更新資料
        + AJAX => URL : accountMaintain/ resetQAccountPwd 更新 / 重置 QAccount 登入密碼
        + AJAX => URL : accountMaintain/ updateUserStatus 更新 QPlay 使用狀態
    - 操作流程
      * [ 新增 QAccount ]
        + 填入以下資料欄位

帳號 : 必填

QAccount 登入密碼 : 必填

編號 : 必填

名稱 : 必填

分機號碼 : 選填

E-Mail : 選填

Domain : 必填

Company : 必填

* + - * + 送出資料之前

驗證必填資料是否為空值

驗證 QAccount 登入密碼 是否符合規則

參考 shop.blade.php 的 passwordRegexp

* + - * + 送出資料之後

Server 端會檢查

帳號是否已經存在

編號是否已經存在

新增失敗

跳出提示訊息

新增成功

重整 bootstrapTable 資料列表

* + - * [ 編輯 QAccount ]
        + 點擊 [ 名稱 ]，開啟編輯視窗
        + 可以更新的資料

帳號

編號

名稱

分機號碼

E-Mail

Domain

Company

* + - * [ 更新 QAccount 登入密碼 ]
        + 點擊 [ Reset ]，開啟編輯視窗
      * [ 重置 QAccount 登入密碼 ]
        + 點擊 [ Reset Origin ]，開啟確認視窗
      * [ 使用狀態 ]
        + 點擊 [ Y / N ]，開啟確認視窗

##### QPlay Backend Platform 手動操作

* + [ 系統管理 ] > [ 選單維護 ] > 點擊 ( **功能管理** ) > [ 新增 子選單 ]
    - 菜單名 : USER\_QACCOUNT\_MAINTAIN
    - 鏈結 : qAccountMaintain
    - 英文名 : QAccount Maintain
    - 簡體中文名 : QAccount 管理
    - 繁體中文名 : QAccount 管理
  + [ 系統管理 ] > [ 群組維護 ]
    - Administrator : 預設有最高權限，可以使用全部群組，包含剛剛新增的 [ QAccount Maintain ]。
  + 上述動作完成之後，必須修改以下檔案，以 (菜單名) 為變數名稱，建立相對應的資料
    - qplay/ resources/ lang/ en/ **messages.php**
    - qplay/ resources/ lang/ zh-cn/ **messages.php**
    - qplay/ resources/ lang/ zh-tw/ **messages.php**
    - 新增變數
      * 'TITLE\_USER\_QACCOUNT\_MAINTAIN' => QAccount Maintain
  + 最後，要看 User 的操作權限設定選項中的 [功能管理] 是否被勾起。
    - 備註 : 就算是 Administrator 群組，個人的操作權限不一定跟群組一樣。

### 4.13 UI 示意圖

* 無

### 4.14 Test Case

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |
|  |  |