|  |  |  |  |
| --- | --- | --- | --- |
| **Subject :** | QPlay Backend Platform | **Latest Ver :** | 1.4.3 |
| **Latest Release Date :** | 2018-08-16 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Change History** | | | | |
| Date | Author | Version | Type | Comment |
| 2018-05-23 | Darren.K.Ti | 1.0 | Function  DB | 功能修改記錄 : 4.1  1. QPlay 後台新增功能 [ 公司管理 ]  2. QPlay Login 流程修改  3. DB `qplay` 新增 Table  4. QPlay Backend Login - Remember Me |
| 2018-06-19 | Darren.K.Ti | 1.1 | Function  DB | 功能修改記錄 : 4.2  1. QPlay 後台新增功能 [ 消費券管理 ]  2. DB `qplay` 新增 QPay 相關 Table  3. QPlay API 新增   1. QPay 相關 API 2. 更新 QAccount Password   4. APP QPlay   1. 整合 QPay 2. 新增功能 [ 更新 QAccount Password ]   5. QPay Web 查詢頁面 |
| 2018-06-25 | Darren.K.Ti | 1.2 | Function DB  API | 功能修改記錄 : 4.3  1. QPlay 後台新增功能 [ Function Management ]  2. DB `qplay` 新增 Function Manage 相關 Table  3. QPlay API 新增 Function Manage 相關 API |
| 2018-07-06 | Darren.K.Ti | 1.3 | Function  API | 功能修改記錄 : 4.4  1. Plugin QLogin 頁面 支援 QAccount  2. QPlay API login 支援 QAccount  3. QPlay API register 支援 QAccount  3. APP QPlay 使用 Function Manage |
| 2018-07-16 | Darren.K.Ti | 1.3.1 | Function  DB  API | 功能修改記錄 : 4.3  1. Function Detail 的權限管理, 要處理 company\_label  2. 新增設定 QAccount 的權限等級 |
| 2018-07-20 | Darren.K.Ti | 1.4 | Function  DB  API | 功能修改記錄 : 4.2  1. 完成 2018-06-25 尚未完成的項目 |
| 2018-07-26 | Darren.K.Ti | 1.4.1 | Function  API | 功能修改記錄 : 4.2  1. QPay Web API : 修正執行流程  2. 依據功能，個別評估時程 |
| 2018-08-16 | Darren.K.Ti | 1.4.2 | Function | 功能修改記錄 : 4.3  4.3.12 程式碼修改記錄 -  1. 3.d. 修改文字 'FUNCTION\_VARIABLE\_NAME'  2. 2.h.ii.4.  Function Name 修正為可以編輯修改  Variable Name 修正為不可編輯修改 |
| 2018-08-16 | Darren.K.Ti | 1.4.3 | API | 功能修改記錄 : 4.4  1. QPlay API login 支援 QAccount  2. QPlay API register 支援 QAccount  3. QPlay API logout 支援 QAccount  4.4.12 程式碼修改記錄 qplayApi/  g. app/ Repositories/ UserRepository.php -  funciton QAccountLogin ()  修改程式執行邏輯，QAccount 的  帳號需要比對 `qp\_user`.emp\_no，另外，  不需要驗證 `qp\_user`.ad\_flag = N  Verify.php / CommonUtil.php 新增 function，透過 emp\_no  取得 User 資料 |

# 目錄

[**目錄**](#_fvukxvw609q9) **3**

[**1.0 系統設計 (System Design)**](#_17qb5z9rsywe) **4**

[**2.0 資料庫設計 (Database Design)**](#_svpws9pqxbdm) **5**

[**3.0 Implementation**](#_l3yah7s6jx9k) **5**

[**4.0 功能修改記錄**](#_y9sxdk33wuqt) **6**

[4.4 [ Plugin QLogin 頁面 支援 QAccount，QPlay API login 支援 QAccount，QPlay API register 支援 QAccount，APP QPlay 使用 Function Manage ]](#_hfvgfgw3nbq9) 6

# 

# 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

## 4.4 [ Plugin QLogin 頁面 支援 QAccount，QPlay API login 支援 QAccount，QPlay API register 支援 QAccount，APP QPlay 使用 Function Manage ]

4.4.1 需求說明

* **Plugin QLogin 頁面 支援 QAccount**
  + 修改登入頁面 UI
    - 第一階層 - 選擇公司
    - 第二階層 - 選擇登入方式
      * 使用桌機帳號
      * 其他
    - 備註 :
      * 如果第一階層選擇 [ 協力廠商 ]，則隱藏第二階層的下拉選單。
* **QPlay API register 支援 QAccount**
  + User 可以選擇 QAccount 完成登入。
  + 回傳的資料必須新增 :
    - `qp\_user`.ad\_flag
    - `qp\_user`.company
* **QPlay API login 支援 QAccount**
  + User 可以選擇 QAccount 完成登入。
  + 回傳的資料必須新增 :
    - `qp\_user`.ad\_flag
    - `qp\_user`.company
* **QPlay API logout 支援 QAccount**
* **APP QPlay 使用 Function Manage**
  + 依據 API getFunctionList 回傳的資料，決定使用者是否可以使用下列功能 :
    - News
    - Event
    - QPay
    - Reset QAccount Password
    - 各個 APP 的使用權限
  + 備註 :
    - APP QPlay 暫時維持使用 API getAppList，之後才修改使用 API getFunctionDetail。
    - 這份 PESPIS 文件不包含 APP QPlay 程式碼修改的規劃。

4.4.2 時程評估

* [ Plugin QLogin 頁面 支援 QAccount ]
  + 預估時程 : ? ~ ? (1 Day)
  + 實際時程 : 1 Day
* [ QPlay API login 支援 QAccount ]
  + 預估時程 : ? ~ ? (2 Day)
  + 實際時程 : 1 Day
* [ QPlay API logout 支援 QAccount ]
  + 預估時程 : ? ~ ? (0.5 hour)
  + 實際時程 : 0.5 hour

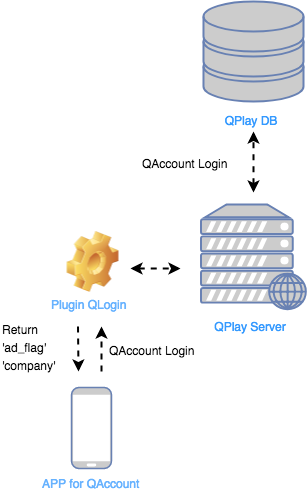
4.4.3 對應文件

* Qplay\_API\_1.3.2\_20180702.docx

4.4.4 修改項目

1. qplayApi/ qplayApi/
2. 修改
   1. app/ Http/ Controllers/ **qplayController.php**
   2. app/ lib/ **CommonUtil.php**
   3. app/ Repositoreies/ **UserRepository.php**
   4. resources/ views/ **login.blade.php**
   5. public/ js/ lang/
      1. **login-en-us.js**
      2. **login-zh-cn.js**
      3. **login-zh-tw.js**
   6. app/ lib/
      1. Verify.php
      2. CommonUtil.php
3. 新增
   1. app/ Services/ **UserService.php**

4.4.5 功能架構圖



4.4.6 資料庫物件清單

* 無

4.4.7 實體關聯圖

* 無

4.4.8 資料表結構

* 無

4.4.9 公用元件

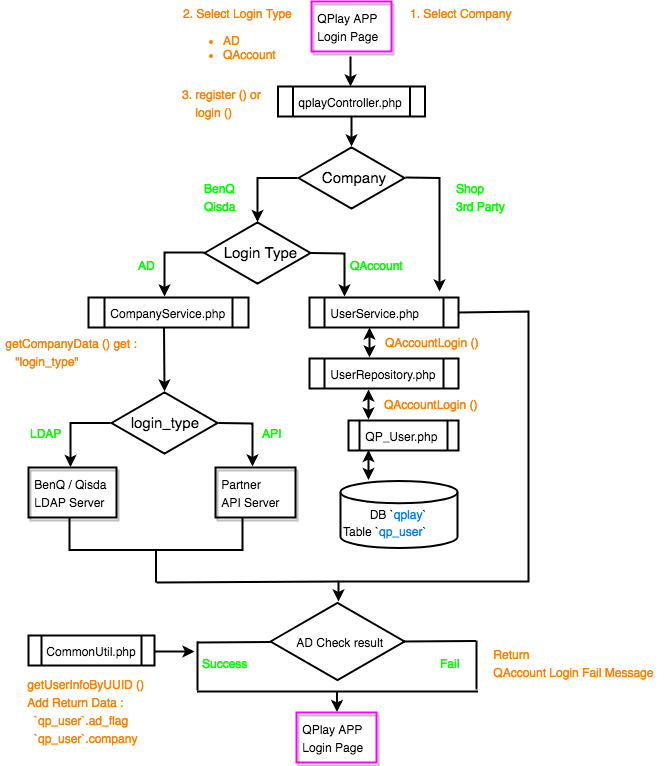
* 無

4.4.10 共用介面

* Qplay\_API\_1.3.2\_20180702.docx

4.4.11 資料流程圖 Data Flow Diagram

* Plugin QLogin 頁面 支援 QAccount
* QPlay API - register 支援 QAccount
* QPlay API - login 支援 QAccount



4.4.12 程式碼修改記錄

* qplayApi/ qplayApi/

1. app/ Http/ Controllers/ **qplayController.php**
   1. 加入 use App\ Services\ UserService.php
   2. function register () 修改執行流程，當以下條件成立時 :
      1. 登入頁面的選項 Company = 協力廠商 (shop)
      2. 登入頁面的選項 Login Type = 其他 (QAccount)
      3. 新增流程，執行 QAccountLogin ()
   3. function login () 修改執行流程，當以下條件成立時 :
      1. 登入頁面的選項 Company = 協力廠商 (shop)
      2. 登入頁面的選項 Login Type = 其他 (QAccount)
      3. 新增流程，執行 QAccountLogin ()
2. app/ lib/ **CommonUtil.php**
   1. 修改 function getUserInfoByUUID () 新增回傳資料 :
      1. `qp\_user`.ad\_flag
      2. `qp\_user`.company
3. resources/ views/ **login.blade.php**
   1. UI 調整 :
      1. 樣式修改
      2. 新增 下拉選單 [ Login Type ]
         1. 使用桌機帳號 (AD)
         2. 其他 (QAccount)
4. public/ js/ lang/
   1. **login-en-us.js**
   2. **login-zh-cn.js**
   3. **login-zh-tw.js**
      1. Plugin QLogin 登入頁面的多國語言文字設定。
      2. 依據新的需求，增加字串。
5. app/ Repositoreies/ **UserRepository.php**
   1. 定義 QAccountLogin ()
      1. 依據 User 在 Login Page 輸入的 Account / Password 進行驗證。
      2. Account 比對 ~~`qp\_user`.loginid~~ `qp\_user`.emp\_no。
      3. Password 比對 `qp\_user`.password。
         1. `qp\_user`.password 的資料預設使用 PHP password\_hash () 進行加密。
         2. Password 必須使用 PHP password\_verify () 進行驗證。
      4. 不需要驗證 `qp\_user`.ad\_flag = N。
      5. 回傳結果。
6. app/ Services/ **UserService.php**
   1. 定義 QAccountLogin ()
      1. 透過 UserRepository.php 進行 QAccountLogin ()，取得回傳結果。
      2. 登入成功 :
         1. 不區分 AD (LDAP / API) 或是 QAccount，只要登入成功，回傳的資料必須新增 :
            1. `qp\_user`.ad\_flag
            2. `qp\_user`.company
         2. 承上，這兩個資料由 CommonUtil::getUserInfoByUUID () 提供。
      3. 登入失敗 :
         1. 當 QAccount 登入失敗的時候，回傳錯誤訊息 :
            1. 以下三個錯誤，個別對應 app/ lib/ ResultCode.php 的代碼

帳號不存在 : \_000901\_userNotExistError

密碼錯誤 : \_000902\_passwordError

帳號停權 : \_000914\_userWithoutRight

1. app/ lib/ Verify.php
   1. 定義 verifyUserByUserEmpNo ()
   2. 定義 verifyUserByUserEmpNo4Logout ()
2. app/ lib/
   1. 定義 getUserInfoByUserEmpNo ()
   2. 定義 getUserInfoByUserEmpNo4Logout ()
   3. 定義 getUserInfoJustByUserEmpNo ()
   4. 定義 getUserStatusByUserEmpNo ()

4.4.13 UI 示意圖

* Plugin QLogin 頁面支援 QAccount

