

智慧入場王

林璟翔 1073522 曾雅詩 1073521 李泳琪 1073531

目錄

一、簡介

1.1 規格目的

1.2 規格範圍

1.3 IEEE 12207.1 6.19.3 (g) 軟體元件間介面定義參考文件

二、規格書內容描述

2.1 系統概述

2.1.1 系統目標

2.1.2 系統範圍

2.1.3 系統架構

2.2 軟體架構

2.3 軟體類別定義

2.4 軟體類別描述

2.5 軟體行為概述

三、軟體設計描述.

3.1 設計概述

3.2 軟體類別設計描述

3.3 執行概念

3.4 需求追溯

四、軟體介面設計描述

4.1 介面設計概述

4.2 外部介面定義

4.3 函式介面定義

4.4 類別介面定義

圖目錄

圖一 系統架構

圖二 Packet diagram

圖三至圖八 Sequence diagram

圖九至圖十五 Collaboration diagram

一、 簡介

1.1 一般性描述

1.1.1 議題日期和狀態：此版於 2021/05/20 之前尚在討論階段未正式完成。

1.1.2 範圍：本規格書的內容以智慧認證系統之類別描述等為主。

1.1.3 議題機構：為了因應疫情，需要快速填寫個人資料所建立之系統。

1.1.4 參考：IEEE 12207.1-1997

1.1.5 內容：智慧認證系統之標準規格書。

1.1.5 主體：包含有系統概述、各項等。

1.1.6 規格書修改歷史：此版於 2021/05/20 為第 1 版。

二、規格書內容描述

2.1 系統概述

2.1.1 系統目標

本系統是為了因應新冠疫情所開發之系統。由於疫情影響，指揮中心公布「實聯制措施指引」。實聯制主要是有疫調需求時，可立即聯繫曾出入同一場所人士，維持國內疫情之穩定控制。本系統為了加速實聯制之填寫時間所開發，主要是用於活動報到現場可使用此系統降低填寫資料的時間，並且主辦方無法獲得使用者個資，只能利用Key 來辨識出有多少參加者，以去除個資外流之疑慮，也可降低在快速填寫資料時意外填入錯誤資料之風險。若該活動有新冠疫情影响者，系統會自動通知該活動主辦方以及參與方。

2.1.2 系統範圍

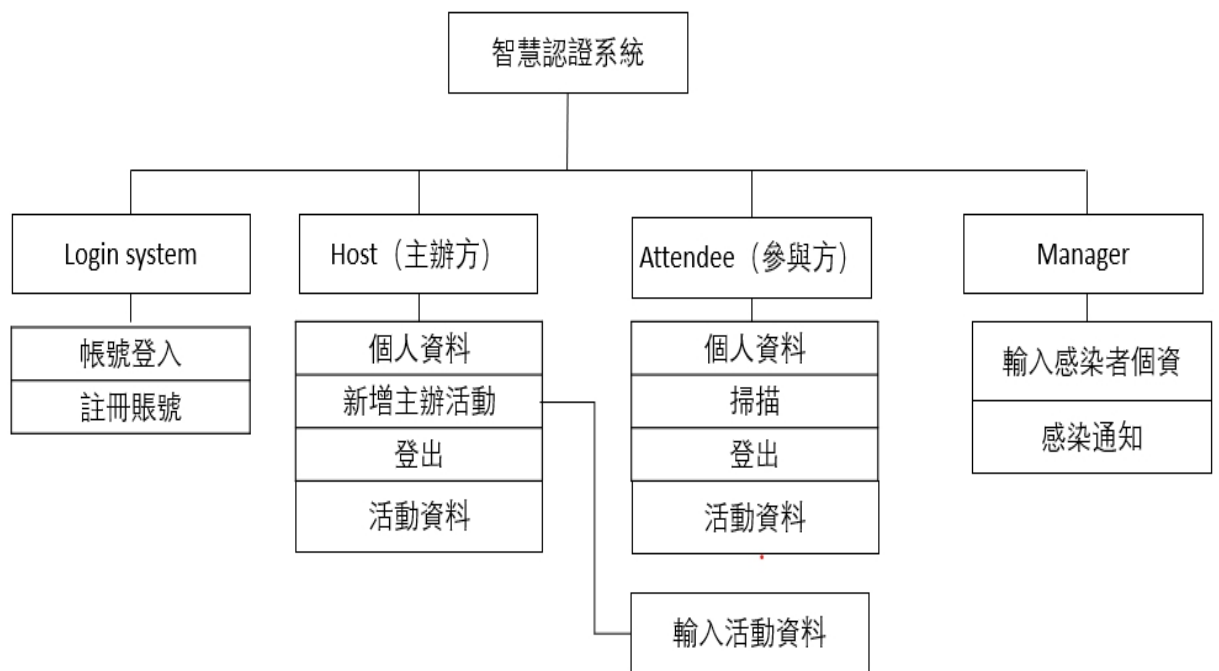
本智慧認證系統之範圍包含主辦方、參與方以及管理員，主辦方、參與方分別對應活動辦理方以及輸入 個資方。管理員之用途為新增新冠病毒感染者，系統會自動搜尋該感染者所參加過的活動，若在感染日 之 28 天內，會自動通知該活動主辦方及參與方。

- 主辦方:包含登入以及管理主辦活動(新增、編輯、管理)介面
- 參與方:包含登入以及管理參與活動介面
- 管理員:包含登入以及管理感染者介面

2.1.3 系統架構

主辦方:可以透過此系統快速統計參與人數，若參加活動中有肺炎感染者，系統會自動通知活動主辦方以及參與方。在個資方面，為避免個資外流，主辦方無法直接讀取到完整的參加者資料。 參與方:將個資輸入系統後，使用者可以藉由系統查詢已經參與的活動。

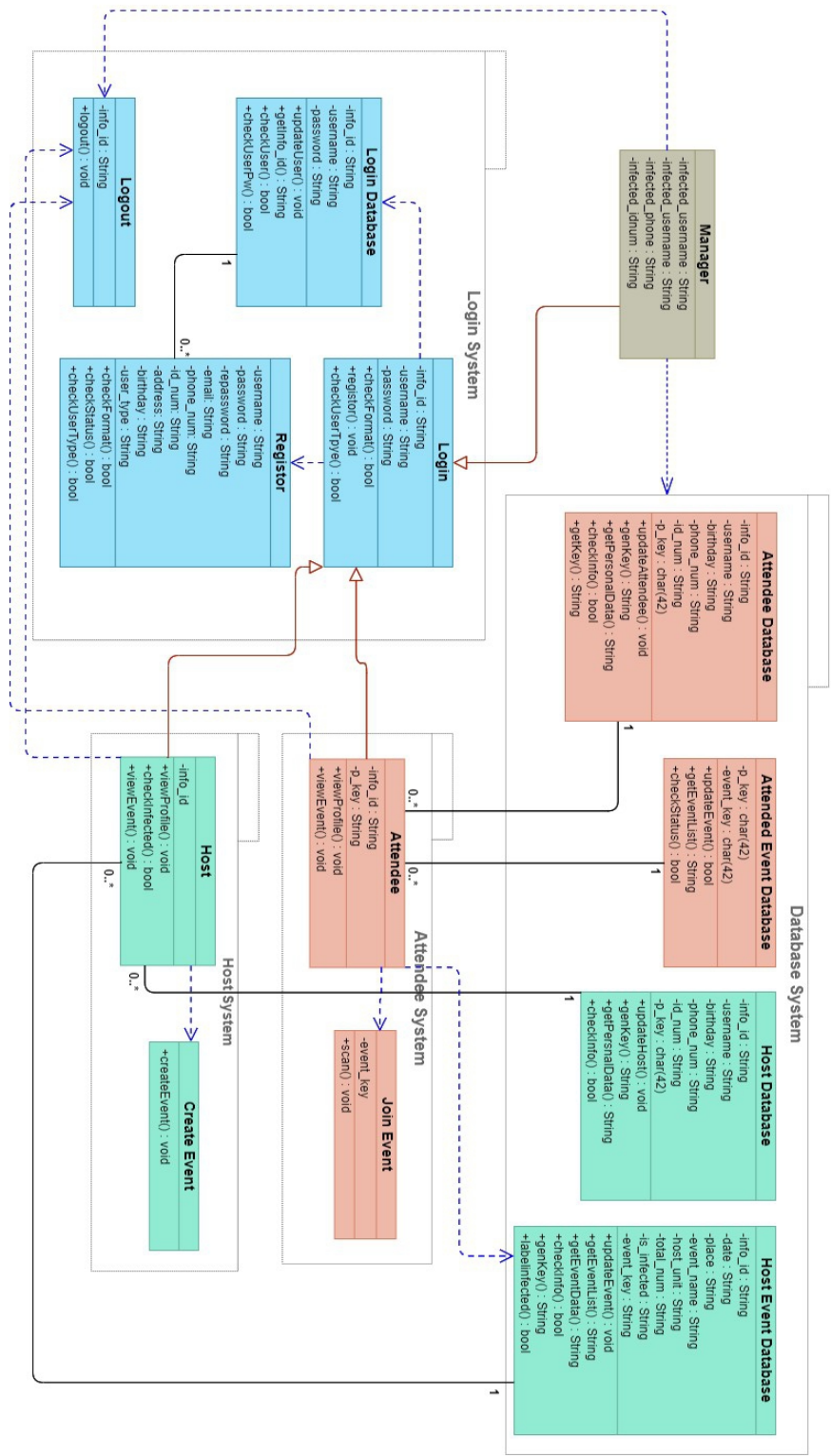
2.2 軟體架構



圖一、系統功能架構圖

- LoginSystem
 - 帳號登入：輸入帳號密碼後登入系統。
 - 註冊帳號：使用者註冊一組帳號，以使用此系統，可以選擇要註冊為主辦方或參與方。
- 主辦方 (Host)
 - 新增主辦活動：新增一個活動。
 - 個人資料：顯示使用者資料。
 - 登出：離開此頁面。
 - 活動資料：顯示主辦活動資料。
- 參與方 (Attendee)
 - 個人資料：顯示使用者資料。
 - 掃描：輸入活動資料。
 - 登出：離開此頁面。
 - 活動資料：顯示已參加活動資料。
- 管理員 (Manager)
 - 輸入感染者：輸入感染者個資。
 - 感染通知：若主辦/參與活動有感染者，會跳出警訊通知。

2.3 軟體類別定義



圖二、Packet Diagram

2.4 軟體類別描述

Class Login
類別描述:
所有使用者皆須登入才能進入系統首頁。
類別屬性:
- private String info_id: 使用者編號 - private String username: 使用者帳號 - private String password: 使用者密碼
類別函式:
+ public bool checkFormat(String username, String password): 檢查輸入格式 + public void register (): 註冊新帳號 + public bool checkUserType(): 檢查使用者類別

Class Register
類別描述:
所有第一次使用本系統的使用者，皆需在首次使用前先註冊。
類別屬性:
- private String username: 使用者帳號 - private String password: 使用者密碼 - private String repassword: 使用者密碼 - private String email: 使用者信箱 - private String phone_num: 使用者手機號碼 - private String id_num: 使用者身份證字號 - private String address: 使用者現居地址 - private String birthday: 使用者出生年月日 - private String user_type: 使用者類別
類別函式:
+ public bool checkFormat(): 檢查輸入格式 + public bool checkStatus(): 檢查註冊狀態 + public bool checkUserType(): 檢查使用者類別

Class Logout
類別描述:
使用者登出系統。
類別屬性:
- private String info_id: 使用者編號
類別函式:
+ public void logout(): 使用者登出

Class Login Database
類別描述:
所有使用者之部分資訊。
類別屬性:
- private String info_id: 使用者編號 - private String username: 使用者帳號 - private String password: 使用者密碼
類別函式:
+ public void updateUser(): 加入資料庫 + public String getInfo_id(): 取得使用者編號 + public bool checkUser(String username): 檢查有無用戶 + public bool checkUserPw(String username, String password): 檢查密碼是否正確

//以上為 Login system

Class Attendee Database
類別描述:
所有參與方使用者之詳細資料。
類別屬性:
<ul style="list-style-type: none"> - private String info_id: 使用者編號 - private String username: 使用者姓名 - private String phone_num: 使用者手機號碼 - private String id_num: 使用者身份證字號 - private String birthday: 使用者出生年月日 - private String p_key: 使用者金鑰
類別函式:
<ul style="list-style-type: none"> + public void updateAttendee(): 加入參與方資料庫 + public String genKey(String info_str): 產生使用者金鑰 + public String getPersonalData(): 取得使用者資料 + public bool checkInfo(): 檢查註冊資訊是否重複 + public String getKey(String infected_idnum): 得到感染者的資料

Class Attendee Event Database
類別描述:
所有已參與活動之詳細資料。
類別屬性:
<ul style="list-style-type: none"> - private String p_key: 使用者金鑰 - private String event_key: 活動金鑰
類別函式:
<ul style="list-style-type: none"> + public void updateEvent(): 加入已參與資料庫 + public String genEventList(String p_key): 取得已參與活動清單 + public bool checkStatus(String p_key, String event_key): 是否加入活動等狀態

Class Host Database
類別描述:
所有主辦方使用者之詳細資料。
類別屬性:
<ul style="list-style-type: none"> - private String info_id: 使用者編號 - private String username: 使用者姓名 - private String phone_num: 使用者手機號碼 - private String id_num: 使用者身份證字號 - private String birthday: 使用者出生年月日 - private String p_key: 使用者金鑰
類別函式:
<ul style="list-style-type: none"> + public void updateHost(): 加入主辦方資料庫 + public String genKey(String info_str): 產生使用者金鑰 + public String getPersonalData(String info_id): 取得使用者資料 + public bool checkInfo(): 檢查資訊

Class Host Event Database
類別描述:
所有已創建活動之詳細資料。
類別屬性:
<ul style="list-style-type: none"> - private String info_id: 使用者編號 - private String date: 活動舉辦日期 - private String place: 活動舉辦地點 - private String event_name: 活動名稱 - private String host_unit: 主辦單位 - private String total_num: 參加人數限制 - private bool is_infected: 有無染疫者參與 - private String event_key: 活動金鑰
類別函式:
<ul style="list-style-type: none"> + public void updateEvent(): 加入活動資料庫 + public String genKey(String info_str): 產生活動金鑰 + public String getEventList(String info_id): 取得已創建活動清單 + public String getEventData(String event_key): 取得該活動資料 + public bool checkInfo(): 檢查資訊 + public bool labelInfected(String p_key): 標記為感染者

Class Attendee
類別描述:
使用者中的參與方，用於取得已參與活動資訊與參加活動。
類別屬性:
<ul style="list-style-type: none"> - private String info_id: 使用者編號 - private String p_key: 使用者金鑰 - private String event_key: 活動金鑰
類別函式:
<ul style="list-style-type: none"> + public void viewProfile(String info_id): 查看個人資料 + public void viewEvent(String event_key): 查看該活動資料

Class Join Event
類別描述:
使用者之參與方加入活動。
類別屬性:
<ul style="list-style-type: none"> - private String event_key: 活動金鑰
類別函式:
<ul style="list-style-type: none"> + public void scan(String event_key): 掃描或輸入活動金鑰

//以上為 attendee system

Class Host
類別描述:
使用者中的主辦方，用於取得已創建活動資訊、創建活動與檢查活動。
類別屬性:
- private String info_id: 使用者編號 - private String event_key: 活動金鑰
類別函式:
+ public void viewProfile(String info_id): 查看個人資料 + public void viewEvent(String event_key): 查看該活動資料 + public bool checkInfected(String event_key): 查看該活動是否有感染者參與

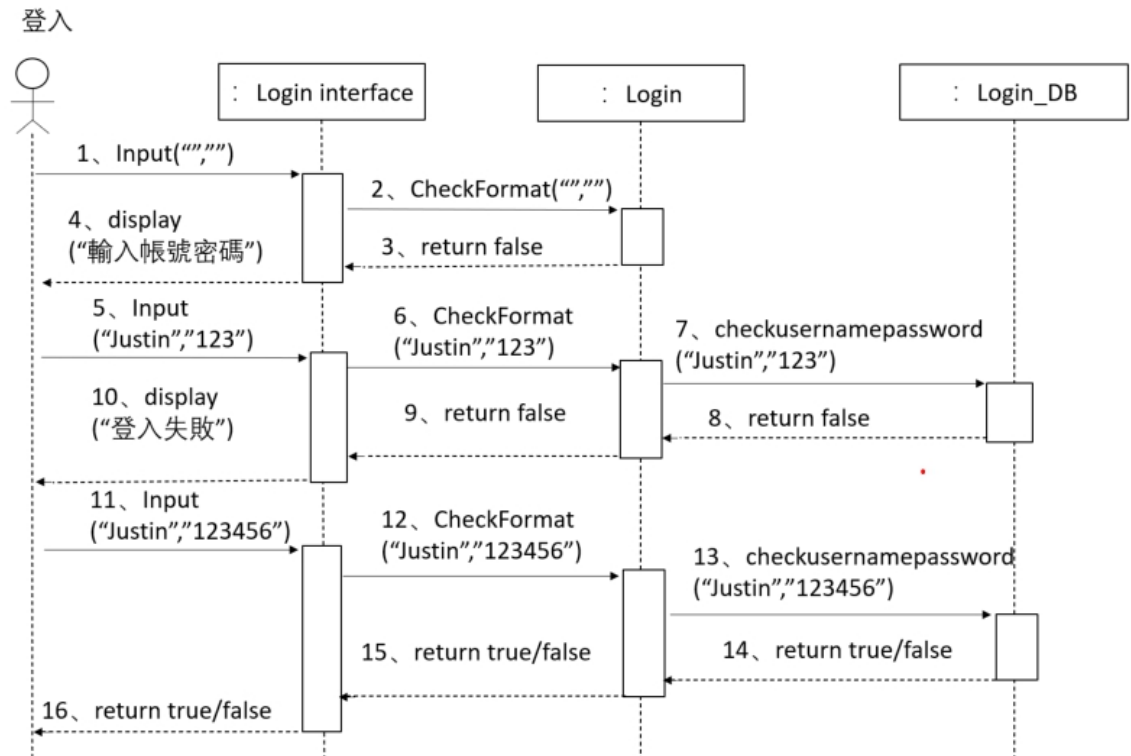
Class Create Event
類別描述:
使用者中的主辦方創建活動。
類別屬性:
- private String info_id: 使用者編號
類別函式:
+ public void createEvent(): 創建活動

// 以上 Host system

Class Manager
類別描述:
使用者中的管理員，得知感染者資訊並更新感染者參與活動等相關動作。
類別屬性:
<ul style="list-style-type: none">- private String infected_username: 感染者帳號- private String infected_birthday: 感染者生日- private String infected_phone: 感染者手機號碼- private String infected_idnum: 感染者身份證字號

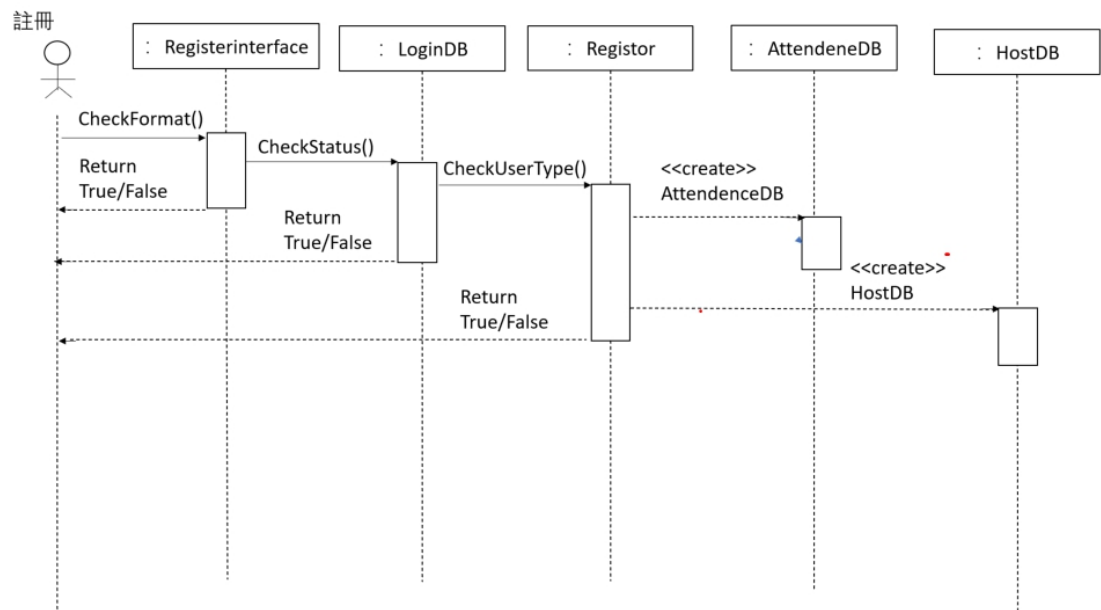
2.5 軟體行為描述

➤ 登入



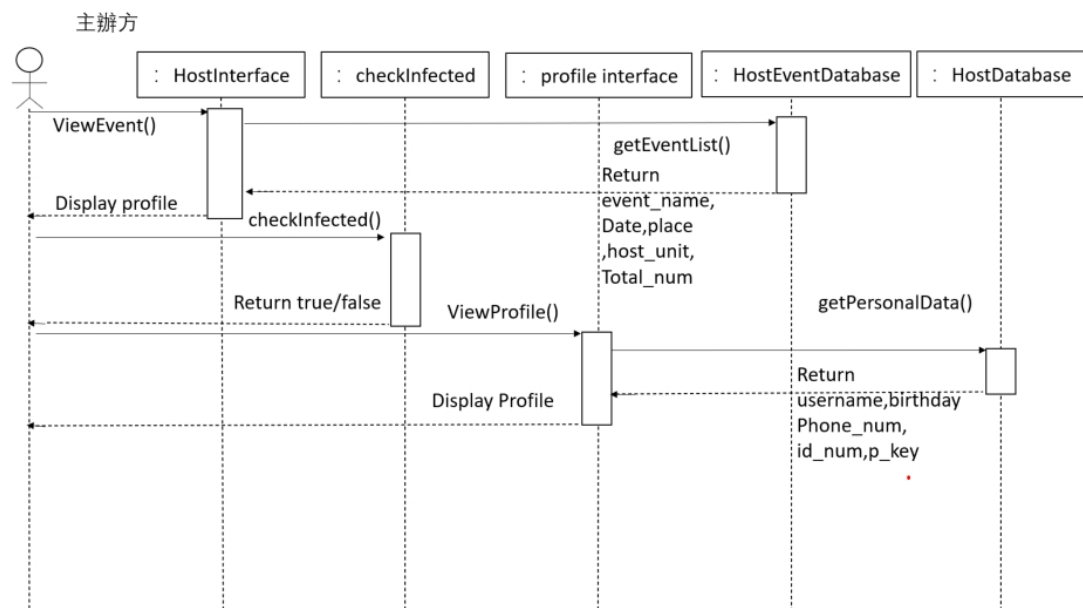
圖三、登入時序圖

➤ 註冊



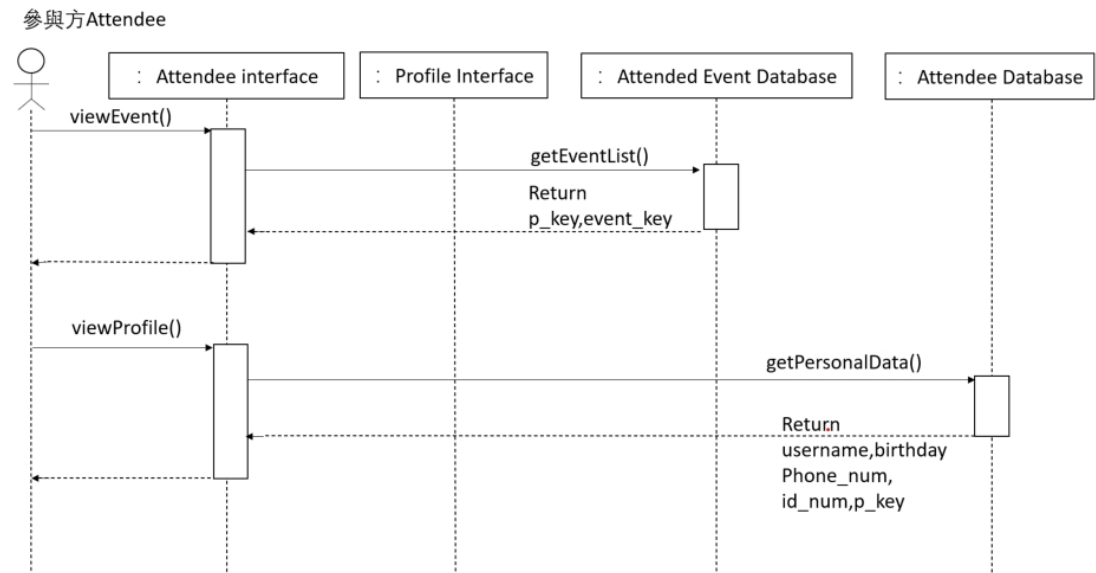
圖四、註冊時序圖

➤ 主辦方



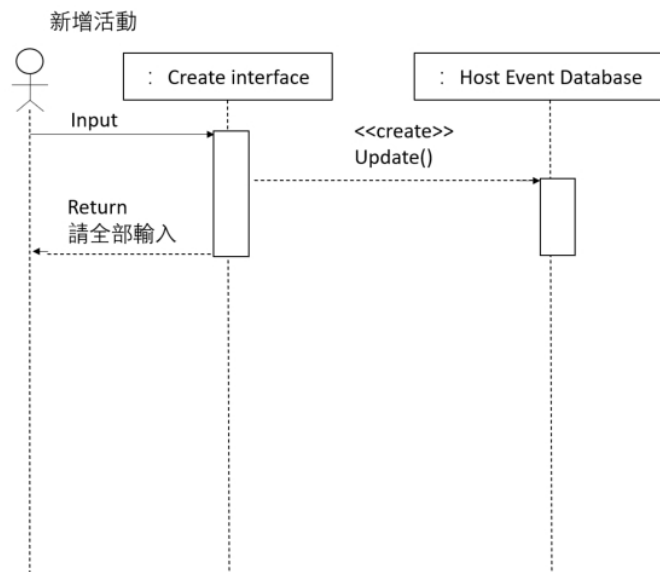
圖五、主辦方時序圖

➤ 參與方



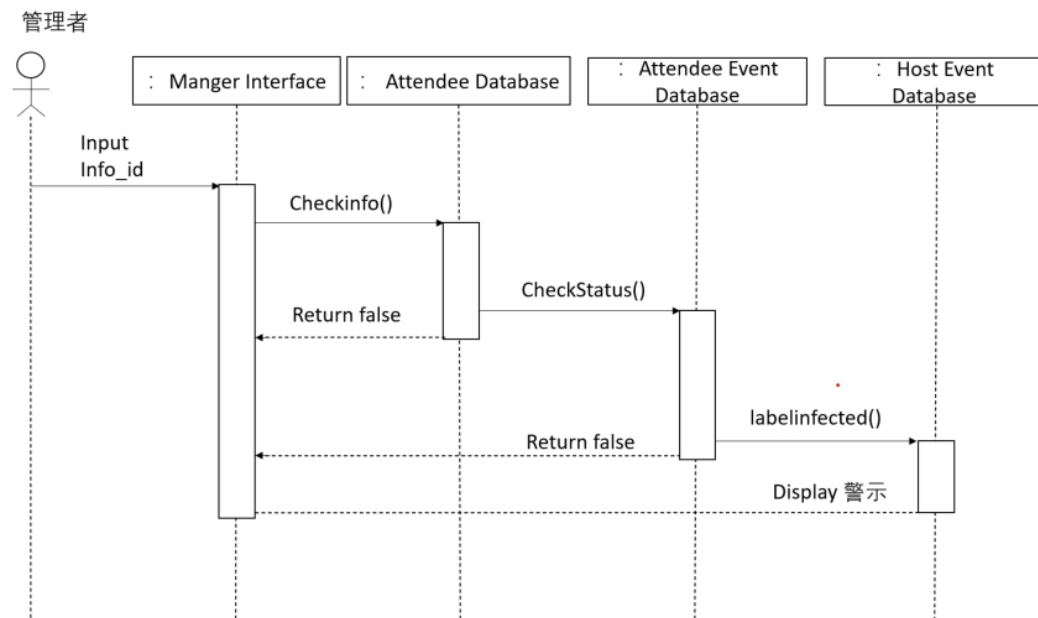
圖六、參與方時序圖

➤ 新增活動



圖七、新增活動時序圖

➤ 管理者



圖八、管理者時序圖

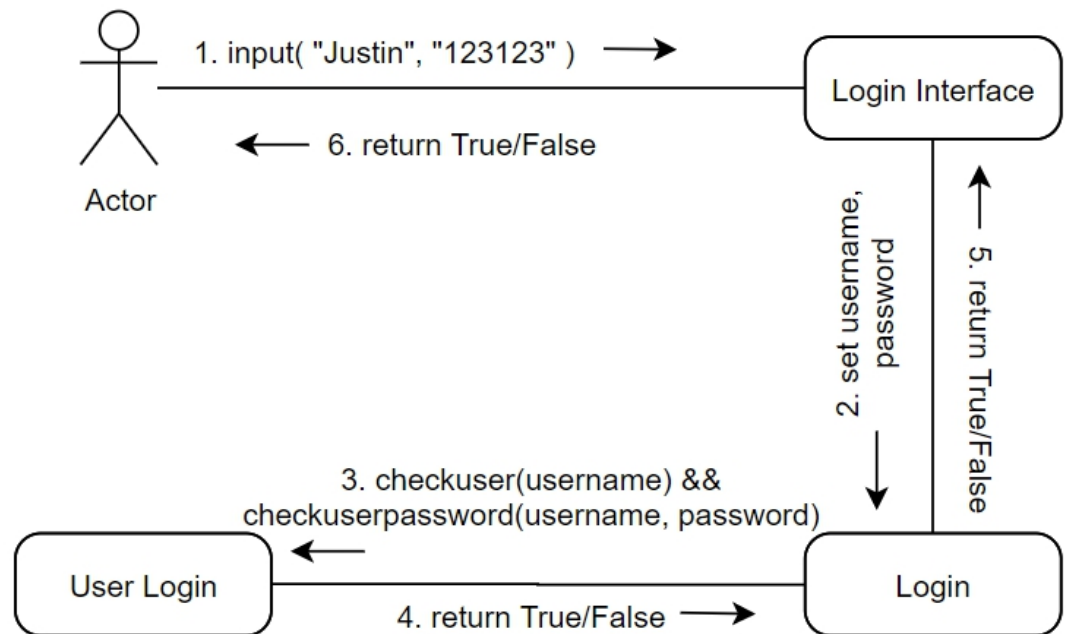
三、軟體設計描述

3.1 設計概述

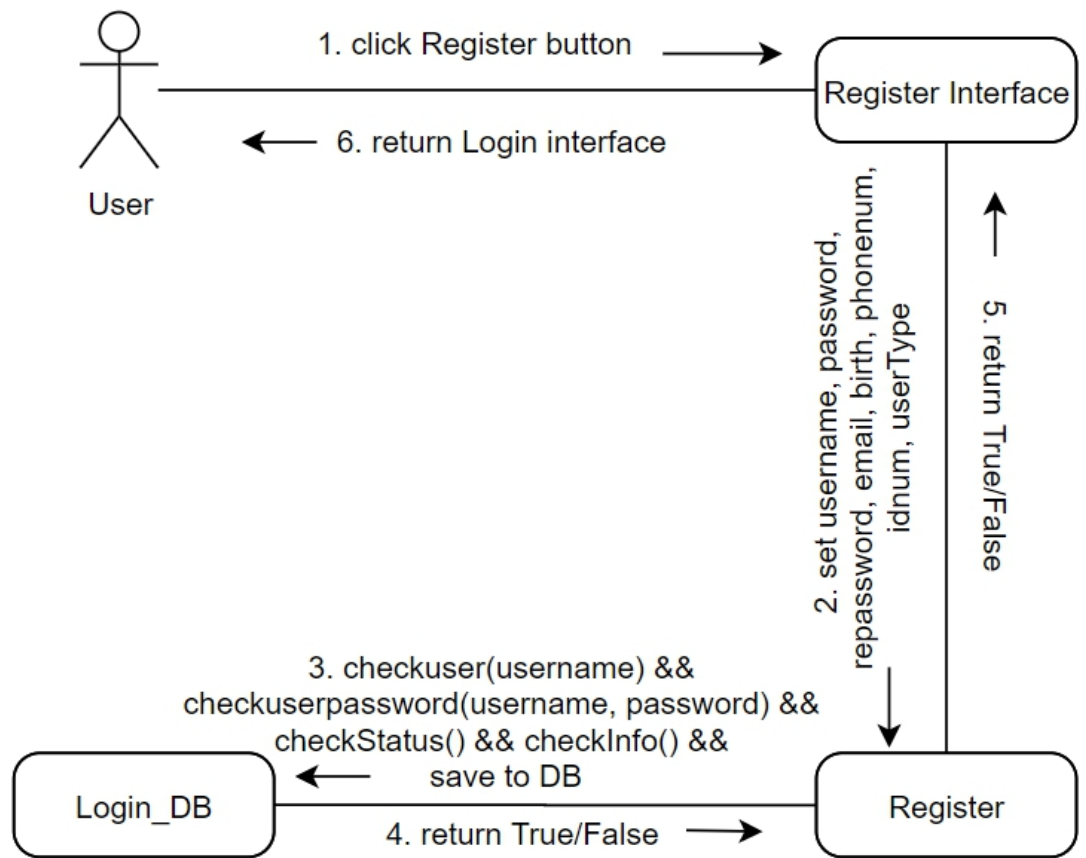
本節內容主要是子系統之間的合作關係、各個子系統內部之類別合作關係及其類別相對應之虛擬碼等系統元件設計相關描述。

3.2 軟體類別設計描述

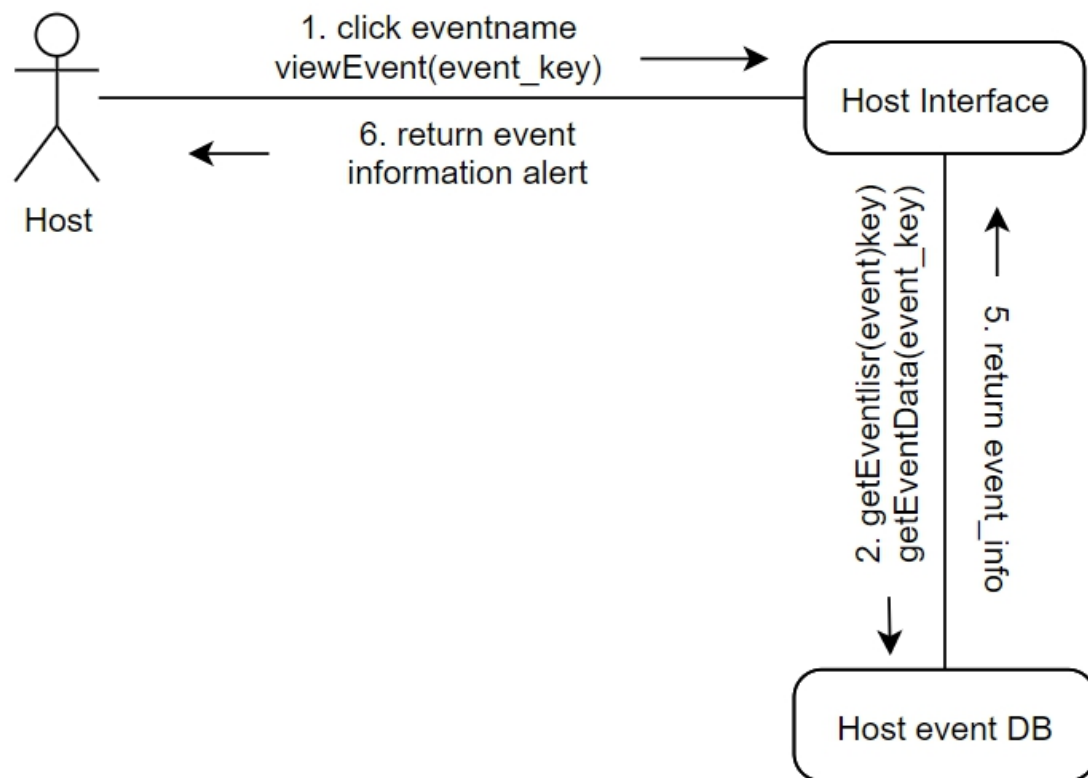
功能包括註冊系統，記錄使用者資料，記錄活動資料，新增活動，新增參加活動，顯示個人資料，顯示活動資料，新增感染者，警告接觸感染者。



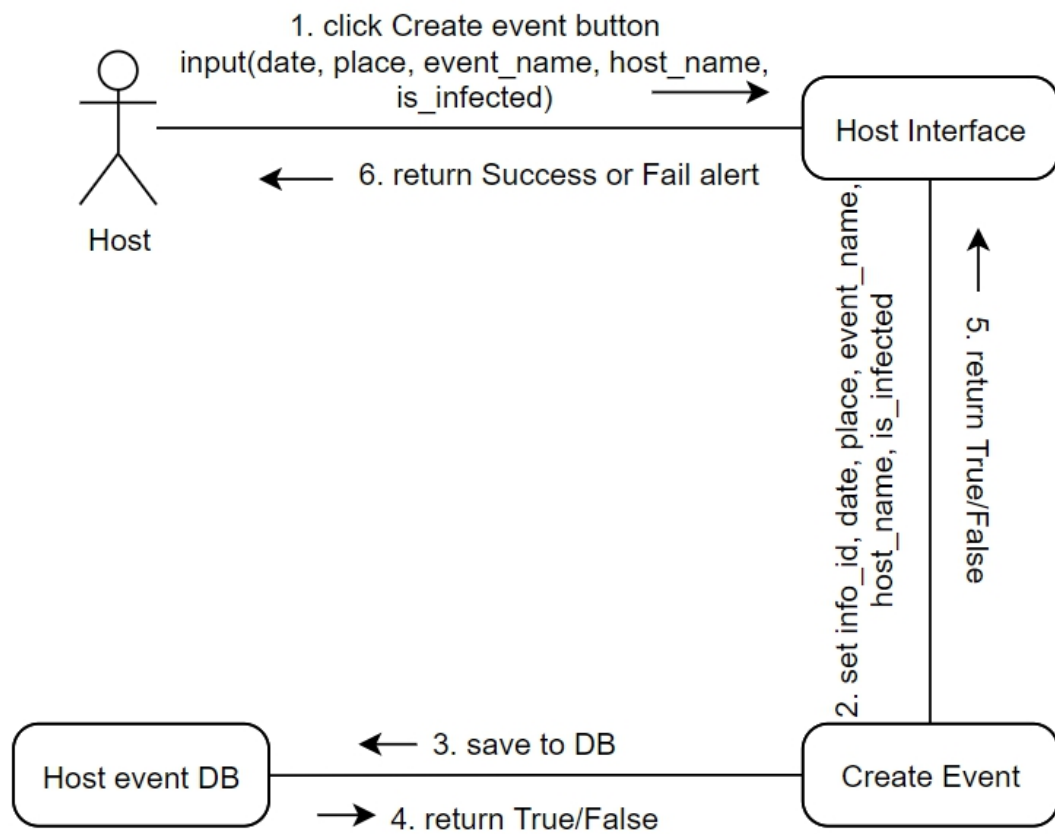
圖九、正常登入



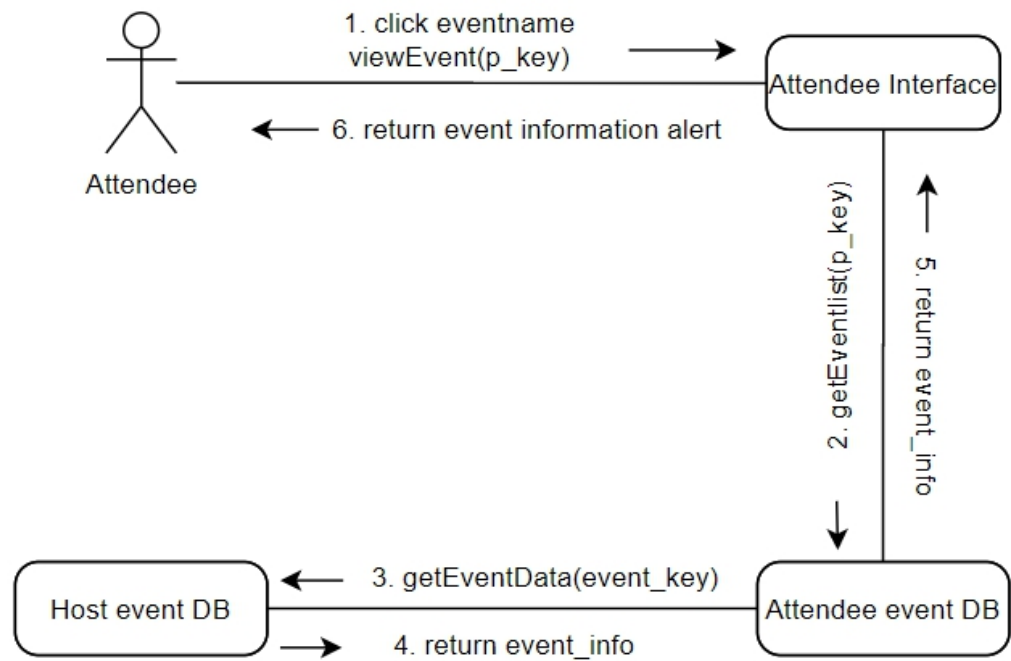
圖十、正常註冊



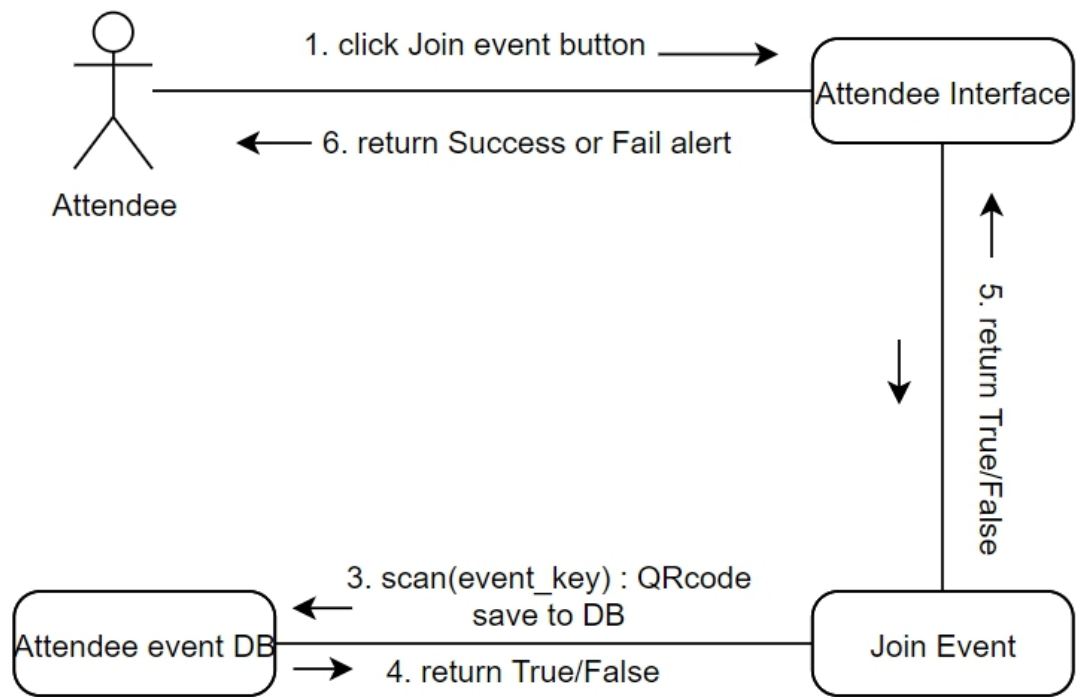
圖十一、主辦方查看活動



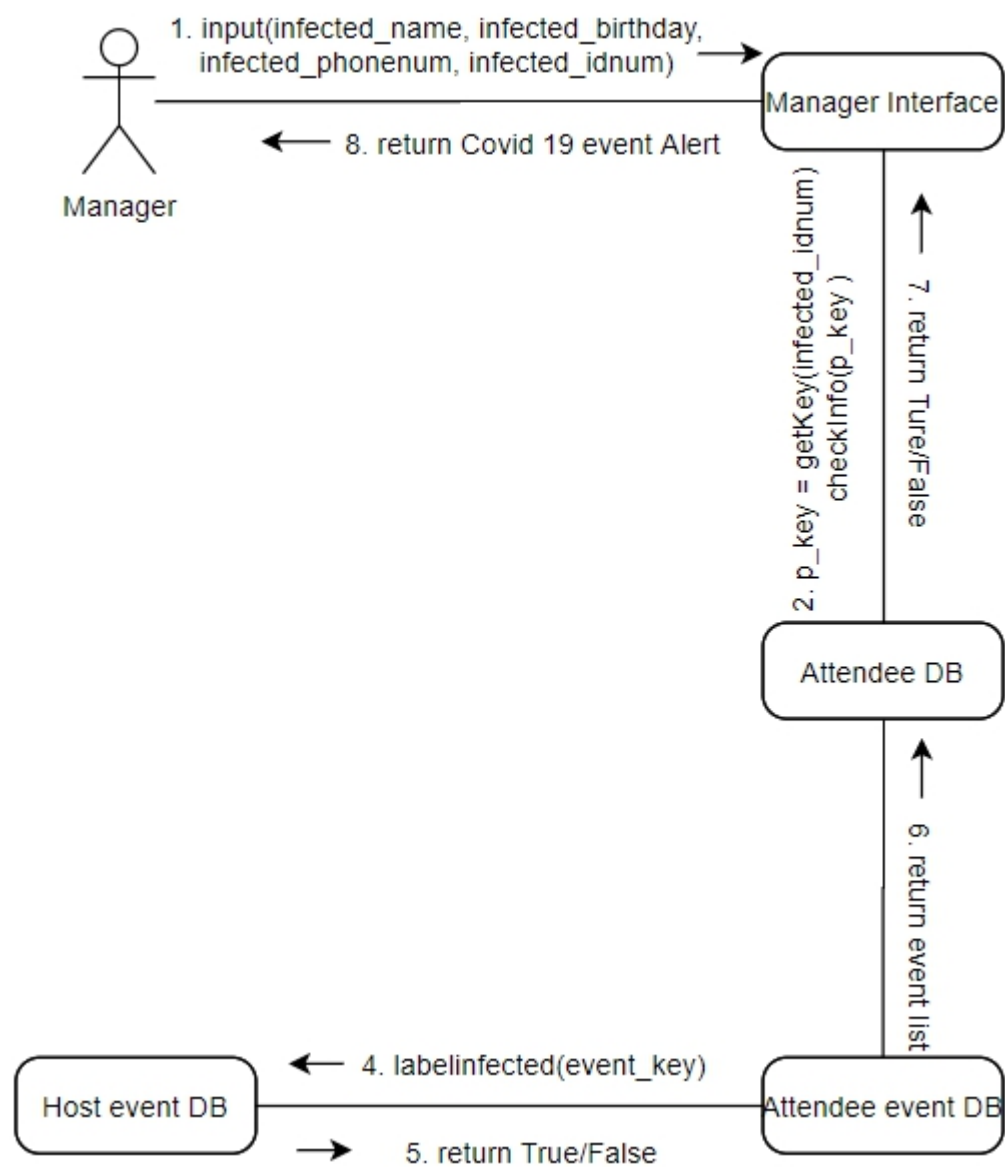
圖十二、主辦方新增活動



圖十三、查看參加活動



圖十四、參加活動



圖十五、管理員

➤ Attendee_Event DB

public Boolean insertData(String p_key, String event_key)

-函式說明：新增 Attendee 參加的活動

-輸入參數：Attendee Key、Event Key

-輸出參數：無

-虛擬碼：

```
{  
    insert Attendee Key、Event Key into Attended_Event DataBase;  
}
```

public String[] attended_event_list(String p_key)

-函式說明：回傳使用者參加過的所有活動

-輸入參數：Attendee Key

-輸出參數：參加過的活動 list

-虛擬碼：

```
{  
    find attended event in Attended_Event DataBase and return;  
}
```

public String is_attended(String p_key, String event_key)

-函式說明：尋找使用者是否參加過該活動，若無新增活動

-輸入參數：Attendee Key、Event Key

-輸出參數：新增活動成功與否

-虛擬碼：

```
{  
    check whether attended event in Attended_Event DataBase;  
    if is attended: return insert fail;  
    else return insert success;  
}
```

public int attended_people(String event_key)

-函式說明：尋找參加過該活動的所有人員

-輸入參數：Event Key

-輸出參數：參加過該活動的人數

-虛擬碼：

```
{  
    find attended people in Attended_Event return;  
}
```

Attendee DB:

public String[] get_personal_data(int id)

-函式說明：尋找 Attendee 的個人資料

-輸入參數：Attendee 的身分證
-輸出參數：Attendee 的個人資料
-虛擬碼：

```
{  
    find Attendee personal data in Attendee DataBase and return;  
}
```

```
public Boolean insertData(String Info_id, String Name, String Birthday, String  
Cellphone, String Personal_id)
```

-函式說明：新增 Attendee 的個人資料
-輸入參數：Attendee 的 Info_id、名字、生日、手機、身分證
-輸出參數：成功與否
-虛擬碼：

```
{  
    use datas to call shaEncrypt and get personal key  
    insert all datas into Attendee DB;  
    if success: return true;  
    else return false;  
}
```

```
public static String shaEncrypt(String strSrc)
```

-函式說明：計算 Attendee 的 key
-輸入參數：Attendee 的 Info_id、Name、Birthday、Cellphone、
Personal_id 組合成的 String
-輸出參數：personal key
-虛擬碼：

```
{  
    calculate personal key and return;  
}
```

```
public Boolean checkinfo(String birth, String phonenum, String identitynum)
```

-函式說明：比對資料庫中是否有相同的個人資料
-輸入參數：生日、手機、身分證
-輸出參數：是與否
-虛擬碼：

```
{  
    check whether datas is in Attendee DataBase or not;  
    if true: return true;  
    else return false;  
}
```

```
public String get_key(String id)
```

函式說明：回傳此身分證的人的個人 key
輸入參數：身分證
輸出參數：Attendee key
-虛擬碼：

```
{
    if id is in Attendee DataBase or not;
    if true: return key;
    else return "error";
}
```

-操作：

系統登入後，顯示 Attendee 頁面。

- 使用"使用者名稱"Button 顯示個人資料
- 使用"登出"Button 回到登入頁面
- 使用"活動名稱"Button 顯示活動資訊
- 使用"參加活動"Button 新增參加活動

Host DB:

```
public String[] get_personal_data(int id)
```

-函式說明：尋找 Host 的個人資料

-輸入參數：身分證

-輸出參數：Attendee key

虛擬碼：

```
{
    find Host personal data in Host DataBase and return;
}
```

```
public Boolean insertData(String Info_id, String Name, String Birthday, String Cellphone,
String Personal_id)
```

-函式說明：新增 Host 的個人資料

-輸入參數：Host 的 Info_id、名字、生日、手機、身分證

-輸出參數：Host key

虛擬碼：

```
{
    use datas to call shaEncrypt and get personal key
    insert all datas into Host DB;
    if success: return true;
    else return false;
}
```

```
public static String shaEncrypt(String strSrc)
```

-函式說明：計算 Host 的 key

-輸入參數：Host 的 Info_id、Name、Birthday、Cellphone、Personal_id 組合成的 String

-輸出參數：personal key

虛擬碼：

```
{
    calculate personal key and return;
}
```

➤ **Hosted_event DB:**

```
public Boolean insertData(String Info_id, String Date, String Place, String Event_name,
String Host_name, String Max_ppl, String is_infected)
```

-函式說明：新增新活動

-輸入參數：Host 的 Info_id，活動日期、時間、地點、活動名稱、主辦方名稱、
參加上限人數、是否有人被感染

-輸出參數：成功與否

虛擬碼：

```
{
    use datas to call shaEncrypt to get event key;
    insert Host Info_id, Date, Place, Event_name, Host_name, Max_ppl, event key,
is_infected into Hosted_event DataBase;
    if success: return true;
    else return false;
}
```

```
public String[] hosted_event(int info_id)
```

-函式說明：查詢主辦過的活動

-輸入參數：Host 的 Info_id

-輸出參數：主辦過的活動 key 列表

虛擬碼：

```
{
    find the key of hosted event and return
}
```

```
public String[] find_event_data(String key)
```

-函式說明：查詢活動資料

-輸入參數：活動 key

-輸出參數：活動資料

虛擬碼：

```
{
    find event information in Hosted database;
    if can find: return event information;
    else return error;
}
```

```
public static String shaEncrypt(String strSrc)
```

-函式說明：計算 event 的 key

-輸入參數：Host 的 Info_id，活動日期、時間、地點、活動名稱、主辦方名稱、-
參加上限人數、是否有人被感染組合成的 String

-輸出參數：event key

虛擬碼：

```
{
    calculate personal key and return;
}
```

public Boolean checkinfo(String event_name,String Host_name,String max_ppl)

-函式說明：查詢將舉辦的活動資料是否被使用過

-輸入參數：活動名稱、主辦名稱、參加人數上限

-輸出參數：是否被用過

虛擬碼：

```
{
    find whether event_name, Host_name,and max_ppl is use in the same event:
    if true return true;
    else return false;
}
```

public boolean labelinfected(String event_key)

-函式說明：將活動標示為被感染

-輸入參數：event key

-輸出參數：無

虛擬碼：

```
{
    use event key to find event and label the is_infected to true;
}
```

-操作：

系統登入後，顯示 Host 頁面。

- 使用"使用者名稱"Button 顯示主辦方資料
- 使用"登出"Button 回到登入頁面
- 使用"活動名稱"Button 顯示活動資訊
- 使用"新增活動"Button 新增主辦活動

Login DB:


```
public Boolean insertData(String info_id, String mailaddr, String password, String acc_type)
```

-函式說明：新增帳號

-輸入參數：使用者的 Info_id、信箱、密碼、帳號類型

-輸出參數：成功與否

虛擬碼：

```
{
    insert user Info_id, mailaddr, password, acc_type into Login DataBase;
    if success: return true;
    else return false;
}
```

```
public int getInfo_id()
```

-函式說明：取得使用者的 Info_id

-輸入參數：無

-輸出參數：使用者的 Info_id

虛擬碼：

```
{
    calculate Info_id and return;
}
```

```
public Boolean checkusername(String mailaddr)
```

-函式說明：查詢註冊用信箱是否被使用過

-輸入參數：註冊信箱

-輸出參數：是否被使用過

虛擬碼：

```
{
    if mailaddr is in Login database: return true
    else return false;
}
```

```
public String[] checkusernamepassword(String mailaddr, String password)
```

-函式說明：判定是否能成功登入

-輸入參數：信箱及密碼

-輸出參數：登入成功或失敗

虛擬碼：

```
{
    if mailaddr and password match the record in Login database: return true;
    else return false;
}
```

-操作：

打該本 APP 後

- 使用"帳號"欄位 輸入信箱
- 使用"密碼"欄位 輸入密碼
- 使用"登入"Button 登入系統
- 使用"註冊"Button 註冊本系統

Manager

-操作：

使用 Manager 帳號密碼進入此頁面後

- 使用"感染者姓名"欄位 輸入感染者姓名
- 使用"感染者生日"欄位 輸入感染者生日
- 使用"感染者手機號碼"欄位 輸入感染者手機號碼
- 使用"感染者身分證字號"欄位 輸入感染者身分證字號
- 使用"輸入感染者"Button 將該感染者參加過活動的 is_infected 標籤改為 true

Register:

-操作：

進入註冊頁面後

- 使用"姓名"欄位 輸入姓名
- 使用"密碼"欄位 輸入密碼
- 使用"重複密碼"欄位 確認密碼
- 使用"生日"欄位 輸入生日
- 使用"信箱"欄位 輸入信箱
- 使用"手機號碼"欄位 輸入手機號碼
- 使用"身分證字號"欄位 輸入身分證字號
- 使用"居住地"欄位 輸入居住地
- 選擇"參與方"欄位 註冊成為參與方
- 選擇"主辦方"欄位 註冊成為主辦方
- 使用"註冊"Button 註冊本系統

New Event Registor:

-操作：

Host 按下"新增活動"Button，進入此頁面後

- 使用"活動名稱"欄位 輸入活動名稱
- 使用"活動日期"欄位 輸入活動日期
- 使用"活動地點"欄位 輸入活動地點
- 使用"舉辦單位"欄位 輸入舉辦單位
- 使用"人數上限"欄位 輸入參加人數上限
- 使用"新增活動"Button 新增活動

3.3 執行概念

如圖三至圖八所示

3.4 需求追溯

OOA 需求	OOD 相關類別
操作介面簡單及便利:	Class(Login) Class(Registor) Class(Attendee) Class(Host) Class(Manager)
系統穩定:	Class(Login) Class(Logout)
功能操作:	Class(Create Event) Class(Join Event) Class(Manager)
系統功能:	Class(Login Database) Class(Host Database) Class(Host Event Database) Class(Attendee Database) Class(Attendee Event Database)

4.1 介面設計概述

本節主要描述主要描述登入註冊系統介面、參與者介面及主辦方介面等三部分。

4.2 外部介面定義

- 系統平台
 - 硬體平台：本軟體設定於安卓手機、或各可使用安卓模擬器之硬體平台上執行。
 - 作業系統：支援 Android5 以上系列（5/6/7/8/9/10/11/12）
- 主畫面

- 硬體平台：本軟體設定於安卓手機之硬體平台上執行
- 視窗標題名稱設為 software_engineering
- 本軟體執行於 Android 環境：螢幕 1080x2160 解析度下，型號 pixel_3a_xl 之模擬器。本軟體之視窗範圍不可超出螢幕顯示範圍。
- 本軟體之視窗規劃如下圖所示

<登入>



<參與者介面>



<主辦方界面>



<管理者介面>

9:13 ⓘ



software_engineering



感染者勿資料

感染者姓名

感染者生日

感染者手機號碼

感染者身分證字號

輸入感染者

4.3 函式介面定義表

Class(Login)
+ bool checkFormat(String username, String password)
+ void register ()
+ bool checkUserType()
Class(Registor)
+ bool checkFormat()
+ bool checkStatus()
+ bool checkUserType()
Class(Login Database)
+void updateUser()
+ String getInfo_id()
+ bool checkUser(String username)
+ bool checkUserPw(String username, String password)
Class(Logout)
-void logout()
Class(Host)
-void viewProfile(String info_id)
-bool checkinfected(String event_key)
-void viewEvent(String event_key)
Class(Create Event)
-void createEvent()
Class(Host Database)
-void updateHost()
-String genKey(String info_str)
-String getPersonalData(String info_id)
-bool checkInfo()
Class(Host Event Database)
-void updateEvent()
-String getEventList(String info_id)
-String getEventData(String event_key)
-bool checkInfo()
-String genKey(String info_str)
-bool labelinfected(String p_key)
Class(Attendee)
-void viewProfile(String info_id)
-void viewEvent(String event_key)
Class(Join Event)
-void scan(String event_key)

Class(Attendee Database)
+ void updateAttendee() + String genKey(String info_str) + String getPersonalData() + bool checkInfo() + String getKey(String infected_idnum)
Class(Attendee Event Database)
+ void updateEvent() + String genEventList(String p_key) + bool checkStatus(String p_key, String event_key)
Class(Manager)