



# 系統規格書 v0.2

## 版本控制

版本	文件撰寫	說明	日期
v0.2.0	劉曉萍 林俊吳	資料分析補述、修正錯誤資料	2002-05-13
v0.1.0	劉曉萍 林俊吳	詳述系統架構規劃	2002-04-30

## 版本位置

[v0.1 2002-04-30](#)

## 最新版本

[v0.2 2002-05-13](#)

## 文件撰寫

林俊吳, jWizard <joseph116@giga.net.tw>

劉曉萍, jWizard <yedda\_liu@yahoo.com>

## Abstract

本專案主要在建立一套可快速建置並滿足業者及消費者需求的售票系統解決方案，內含票卷服務資源管理、(實體&線上)售票管理、線上會員管理、線上付費、帳務管理、報表、權限管理…等功能。

## Status of this Document

本文件主要做為jxTicket售票系統建置準則。

第一階段建置時間：2001-05-01 ~ 2001-05-20

## 目錄

<b>1 系統簡介.....</b>	<b>6</b>
1.1 特色.....	6
1.2 限制.....	6
<b>2 系統架構.....</b>	<b>7</b>
2.1 基本架構.....	7
2.1.1 實體架構.....	7
2.1.2 實作架構.....	8
2.2 功能樹狀圖.....	8
2.3 功能模組圖.....	9
2.3.1 Merchant.....	9
2.3.2 Web Plus.....	10
2.4 模組互動圖.....	11
2.5 系統擴充圖.....	13
<b>3 系統分析.....</b>	<b>15</b>
3.1 使用案例.....	15
3.1.1 Merchant系統管理者.....	15
3.1.2 商店服務員.....	16
3.1.3 Web Plus系統管理者.....	17
3.1.4 線上使用者.....	18
3.2 類別關係圖.....	19
3.2.1 票卷服務設定.....	19
3.2.2 線上系統管理.....	20
3.2.3 商店購票.....	21
3.2.4 線上購票.....	21
3.2.5 身份驗證及票卷印出.....	22
3.2.6 退款.....	23
3.3 活動圖.....	24
3.3.1 服務設定.....	24
3.3.2 商店購票及取票.....	25
3.3.3 線上購票及取票.....	26
3.3.4 退款.....	28
<b>4 資料分析.....</b>	<b>29</b>
4.1 資料庫.....	29
4.1.1 Merchant系統資料庫.....	29
4.1.2 Web Plus線上交易資料庫.....	31

---

4.2 XML訊息 .....	32
4.2.1 Architecture.....	32
4.2.2 Resource.....	35
4.2.3 Package.....	36
<b>A 附錄 .....</b>	<b>37</b>
A.1 票卷種類 .....	37
A.2 取票方式 .....	37
<b>B 附件 .....</b>	<b>38</b>
B.1 XML DESIGN - ARCHITECTURE .....	38
B.2 XML DESIGN - RESOURCE.....	42
B.3 XML DESIGN - PACKAGE .....	44
<b>C DB SCHEMA說明 .....</b>	<b>46</b>
C.1 MERCHANT.....	46
C.2 WEBPLUS.....	52
<b>D 系統開發基礎 .....</b>	<b>55</b>
D.1使用技術.....	55
D.2作業環境.....	55
<b>E 系統開發PACKAGE分類 .....</b>	<b>56</b>
E.1 MERCHANT端.....	56
E.2 WEBPLUS端.....	58

## 1 系統簡介

在現今一切講求「快速、有效率」的時代，「排隊」不再被消費者視為理所當然，無法解決「排隊」問題的服務提供者也將因此而減少營收。許多消費服務多因此藉由電話、Internet擴充銷售管道以解決「排隊」問題，尤其是每逢週末假日即容易大排長龍的售票、訂位服務。

綜觀目前業者所提供的售票、訂位服務，可分人工、語音及線上三種方式。人工及語音對消費者來說不夠直覺，消費者很難掌握消費結果並滿足需求；線上售票雖可以快速處理消費者需求並提供消費結果，但現行線上、實體售票系統無法完全整合，票卷資源無法彈性分配，也常造成業者的利潤損失；而且大多數系統多為過份客製化，致使業者異業結盟的擴充彈性小，相對系統的建置時間拉長，不符經濟效益。

### 1.1 特色

- 實體及線上銷售資源不再僵化，未售出的資源可以隨時重新分配
- 可支援任何販賣空間或時間的服務，如戲院、藝文活動、餐廳訂位、交通運輸…等
- 消費者可以不受空間的限制隨時在實體或線上通路預購任何未出售的服務
- 每一項資源單位，如一個座位、房間，不會在任何時間內經由實體通路或線上通路被重覆購買
- 系統建置快速、更具彈性，可配合現有商業模式建構為實體、線上、單一商店、加盟連鎖商店等應用系統
- 對於線上售票應用，可依服務特性彈性選擇發出實體票卷或虛擬票卷
- 以XML做為資料儲存及傳遞的基本格式，其標準化、極具擴充性的特質將反應在系統設計上
- 可輕易地與支援XML、Web Service的外部系統做資料整合

### 1.2 限制

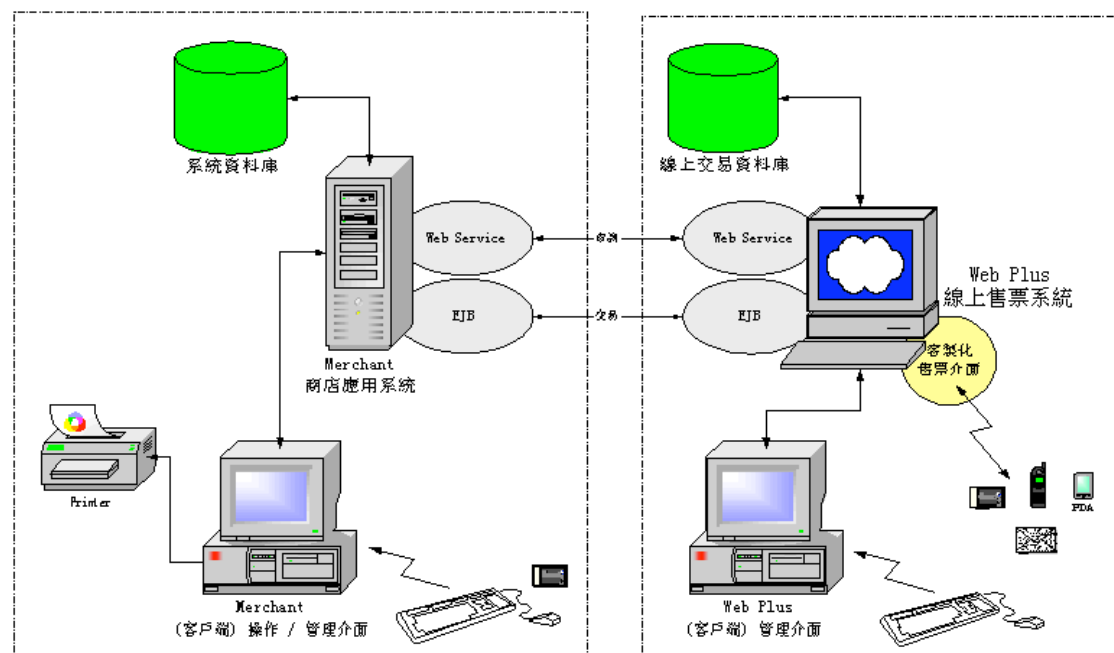
- 每一個實體商店只能為一個線上商店提供票卷資源及售票服務
- 線上訂位無票卷者的身份認證，因實體設備無法配合而未實作相關認證功能，未來可以配合硬體設備以『信用卡』、『IC卡』、『會員卡』等方式做身份確認
- 因刷卡設備無法配合，本系統現階段將實體通路的金流作業(如信用卡刷卡作業)視為獨立運作系統，相關處理系統需由業者自行建構，本系統只出具報表做為對帳之用。
- 由於實體設備無法取得且技術層面差異較大，本系統未定義與語音(IVR)系統整合的整合方案

## 2 系統架構

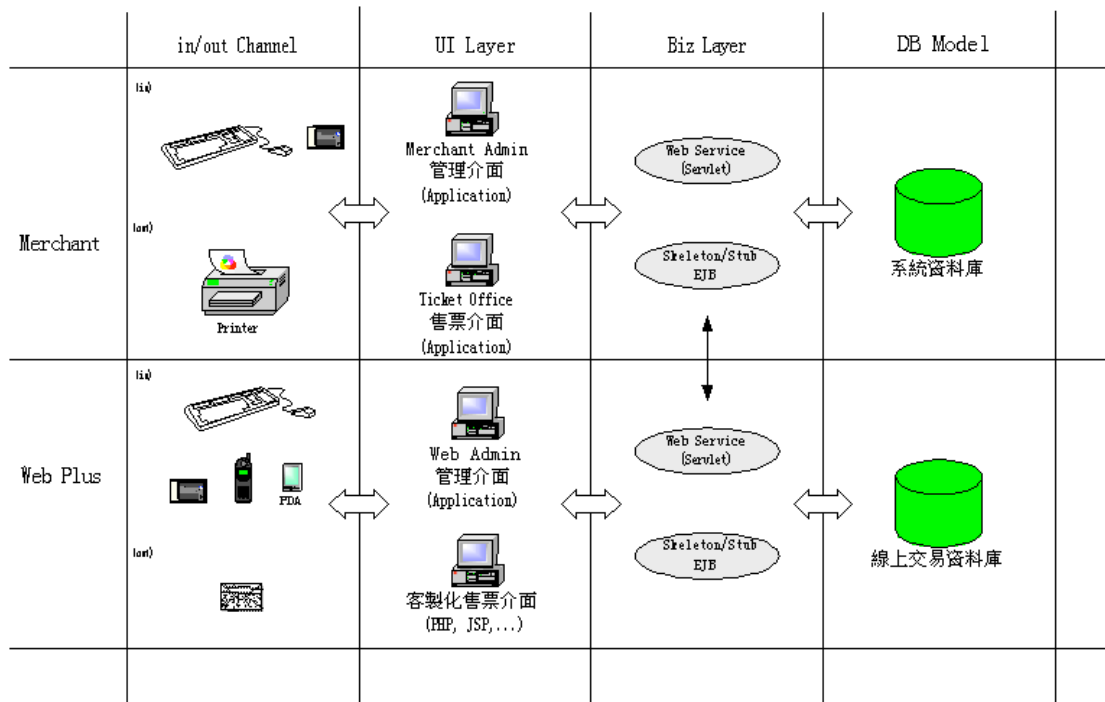
### 2.1 基本架構

系統規劃以「實體通路」為主、「線上通路」為輔，主要提供『Merchant』、『Web Plus』二大應用模組，可依實際商業規模彈性整合成實體、線上、單一商店、加盟連鎖商店等應用模式。

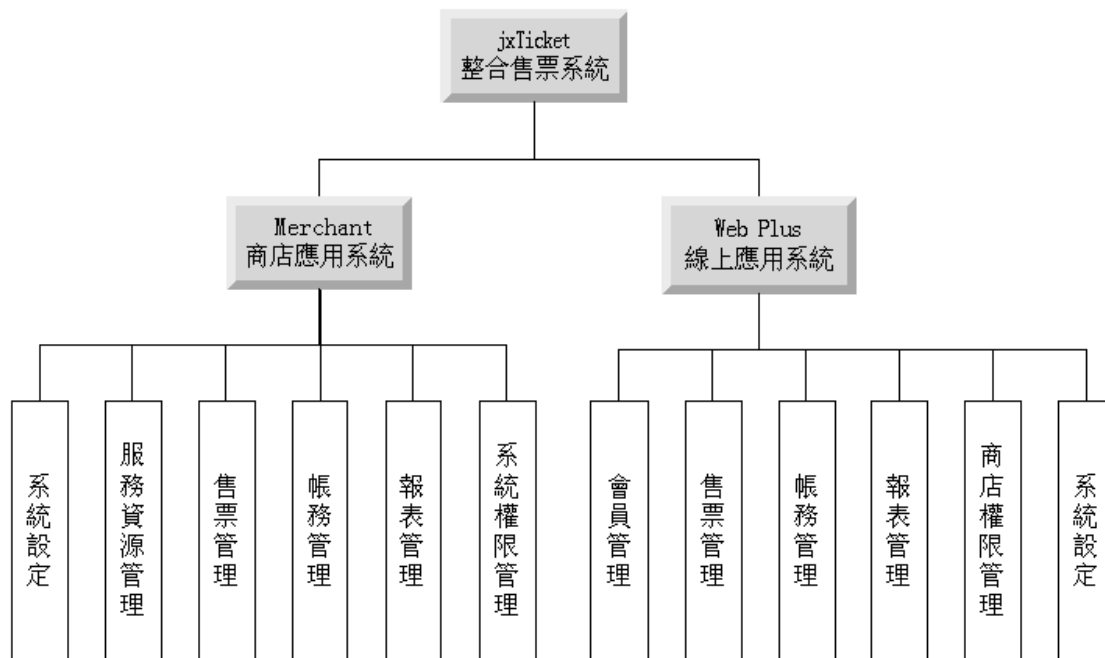
#### 2.1.1 實體架構



## 2.1.2 實作架構

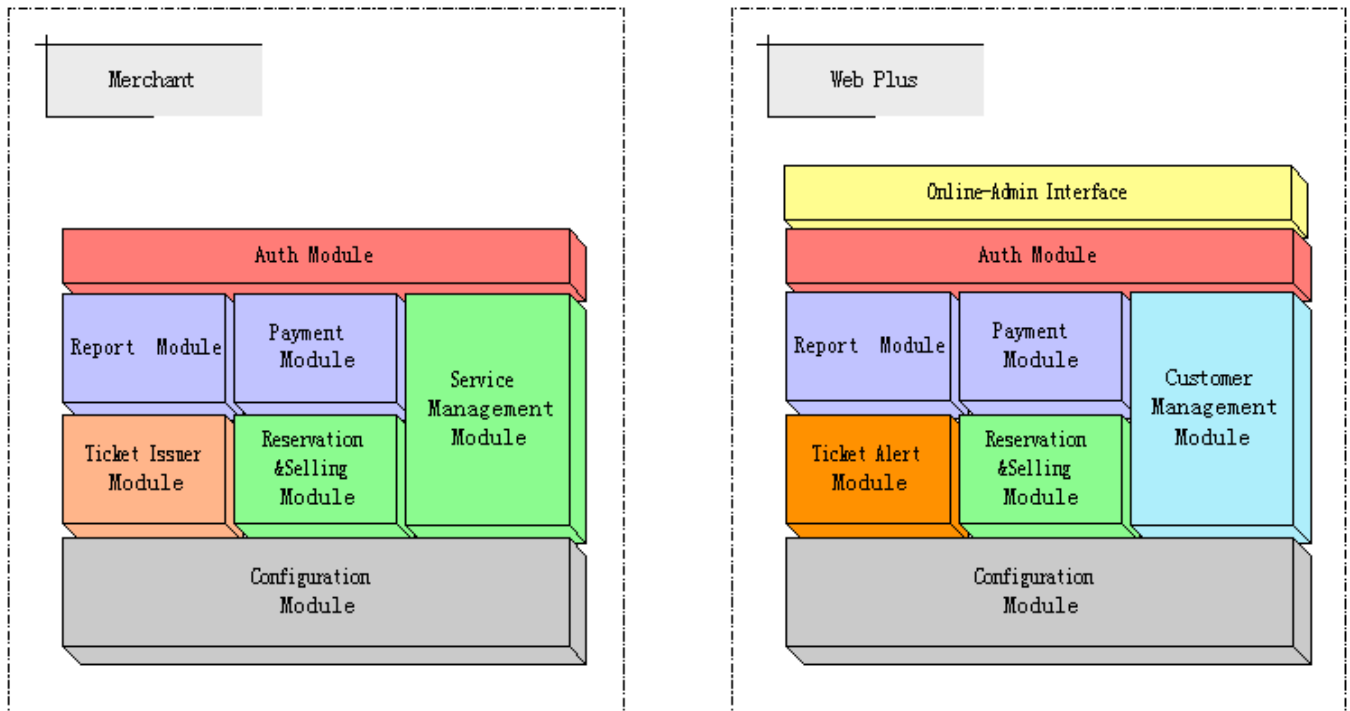


## 2.2 功能樹狀圖





## 2.3 功能模組圖



### 2.3.1 Merchant

- **Auth Module**

處理商店系統權限設定及身份驗證

- **Service Management Module**

處理票卷服務設定及管理

- **Reservation and Selling Module**

負責即時售票、訂位，可接收外接式票卷或訂位驗證工具所傳訊息

- **Ticket Issuer Module**

依設定規格印出票卷

- **Payment Module**

銷售帳務管理

- **Report Module**

處理帳務報表

- **Configuration Module**

負責系統設定

### 2.3.2 Web Plus

- **Auth Module**

處理線上售票系統權限設定及身份驗證

- **Customer Management Module**

負責線上會員資料，做為上購票身份驗證資料來源，同時儲存會員消費記錄

- **Reservation and Selling Module**

負責線上即時售票、訂位

- **Ticket Alert Module**

發送線上訂票確認訊息及傳送印出票卷訊息給Merchant  
Ticket Issuer Module

- **Payment Module**

線上銷售帳務管理，內建SSL線上刷卡授權機制

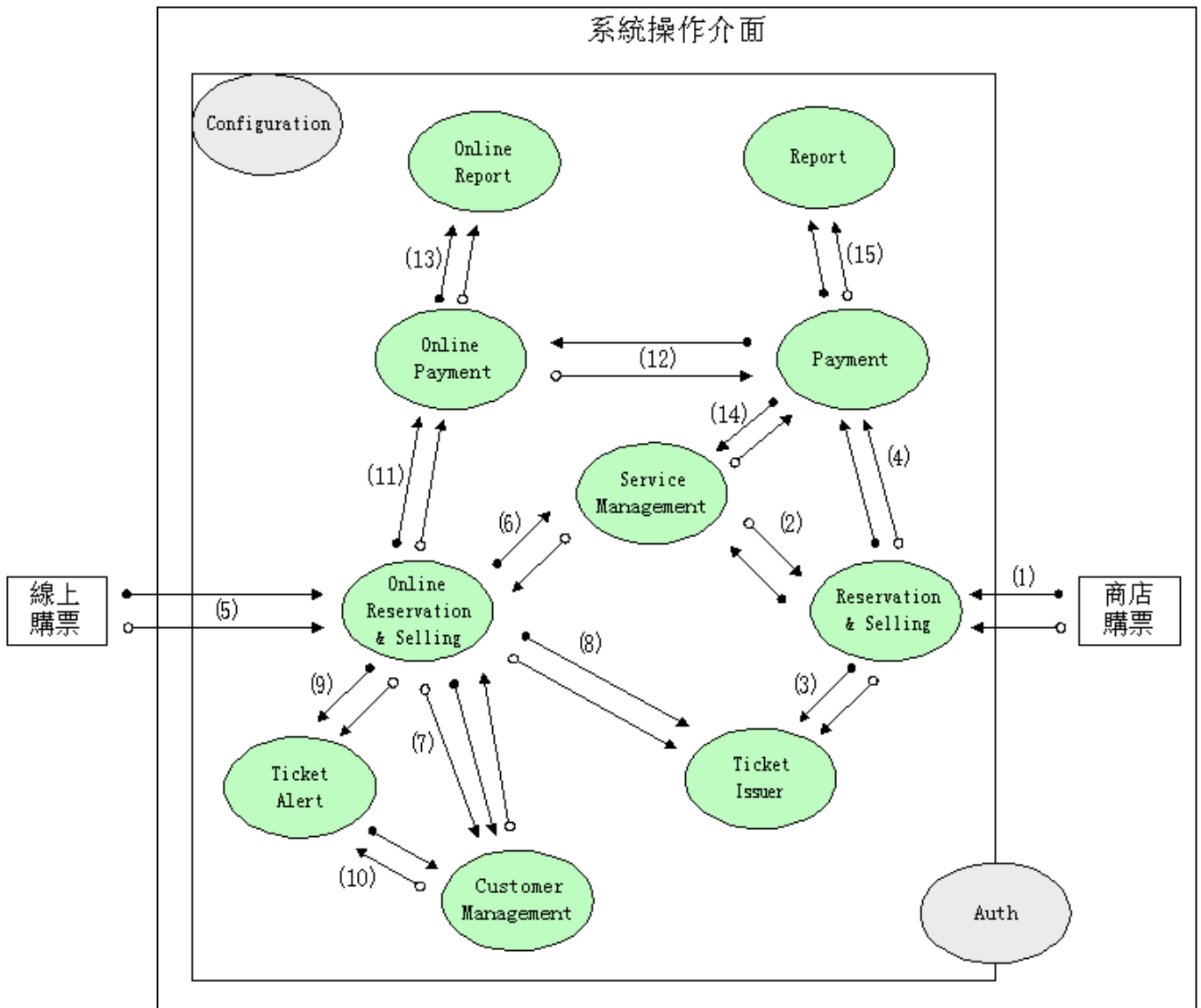
- **Report Module**

處理線上售票帳務報表

- **Configuration Module**

負責系統設定

## 2.4 模組互動圖

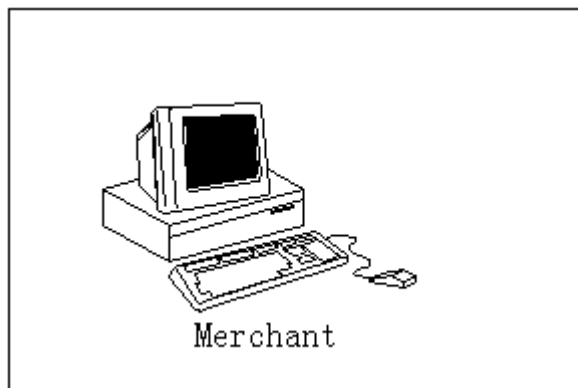


1. Merchant系統操作者選擇票卷服務，執行訂、購票功能
2. Reservation & Selling模組透過Service Management模組取得所有可售票卷資訊
3. 完成購票後，由Reservation & Selling模組傳送交易記錄至Ticket Issuer模組印出票卷
4. Reservation & Selling模組傳送票卷銷售資料至Payment模組處理、儲存；同時Payment模組也負責與外部付款機制溝通，如信用卡、儲值卡付費
5. Web線上使用者選擇票卷服務，執行訂、購票功能

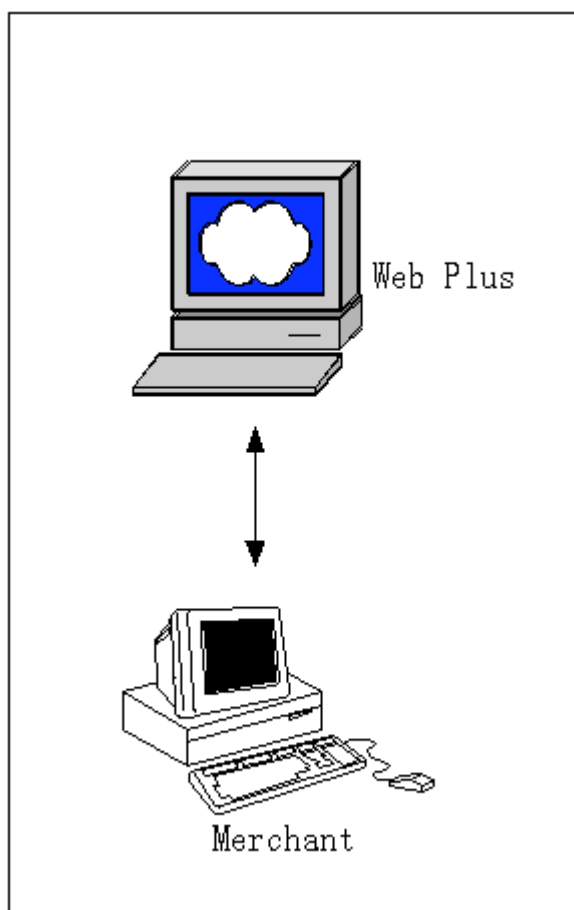
6. Online Reservation & Selling模組透過Merchant的Service Management模組取得所有可售票卷資訊
7. Online Reservation & Selling模組透過Customer Service模組取得線上會員資料以進行線上訂、購票；完成購票後，由Online Reservation & Selling模組傳送該會員交易記錄給Customer Service模組處理、儲存
8. 如使用者選擇郵寄票卷，完成購票後，則由Online Reservation & Selling模組傳送會員資料及交易記錄並至Merchant的Ticket Issuer模組印出票卷
9. 線上會員完成線上訂、購票後，由Ticket Alert模組依會員所設定訊息通知模式事先傳送提醒訊息
10. 傳送提醒訊息前，Ticket Alert模組透過Customer Management模組取得會員資料
11. Online Reservation & Selling模組傳送票卷線上銷售資料至Online Payment模組處理、儲存；同時Online Payment模組也負責與外部付款機制溝通，如信用卡、儲值卡付費
12. Online Report模組定時透過Online Payment模組取得銷售資料以製作銷售統計報表
13. Merchant的Payment模組定期透過Online Payment模組取得線上票卷銷售資料以進行帳務核對
14. 製作報表前由Payment模組透過Service Management模組取得服務銷售基本資料以做為統計基礎
15. Report模組定時透過Payment模組取得所有 (Merchant & Web Plus) 銷售資料以製作銷售統計報表

## 2.5 系統擴充圖

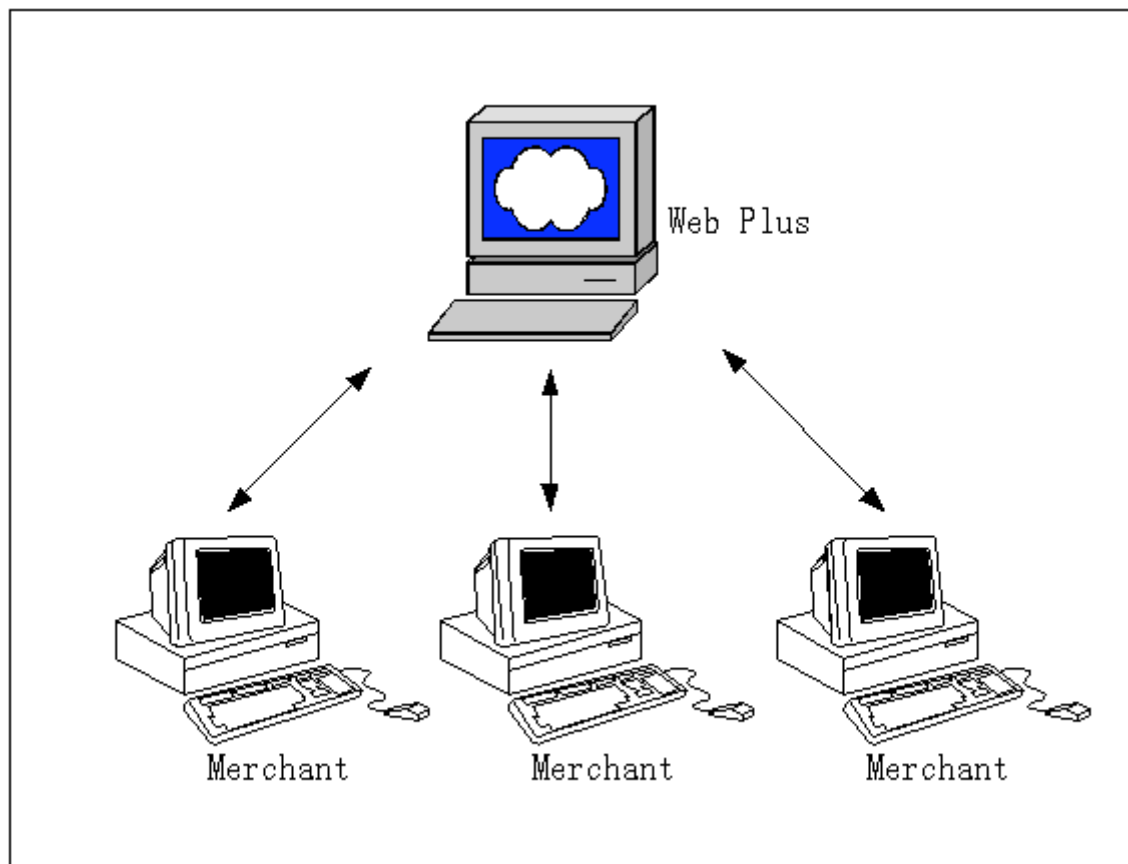
- 單一商店



- 單一商店：線上&實體通路



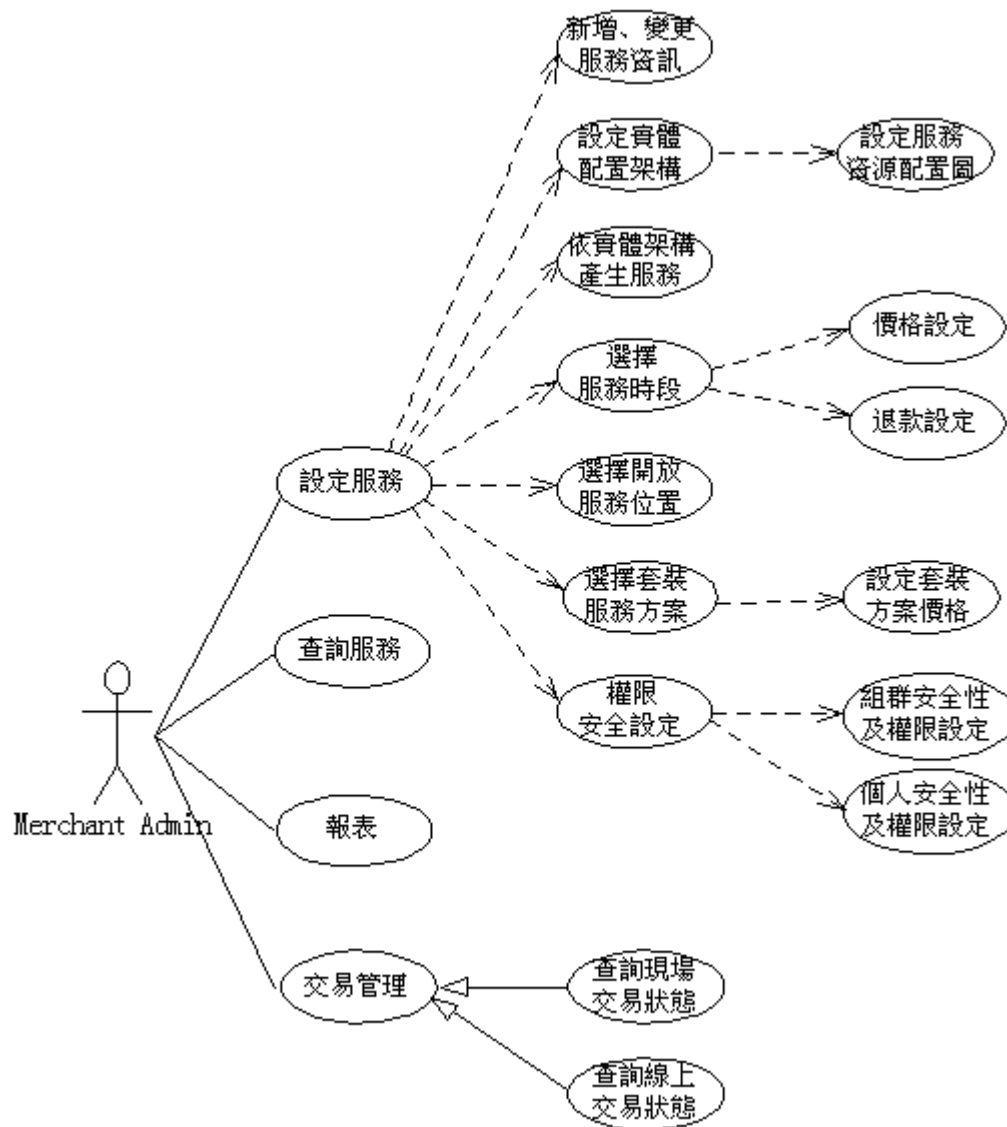
- 連鎖及異業結盟：線上&實體通路



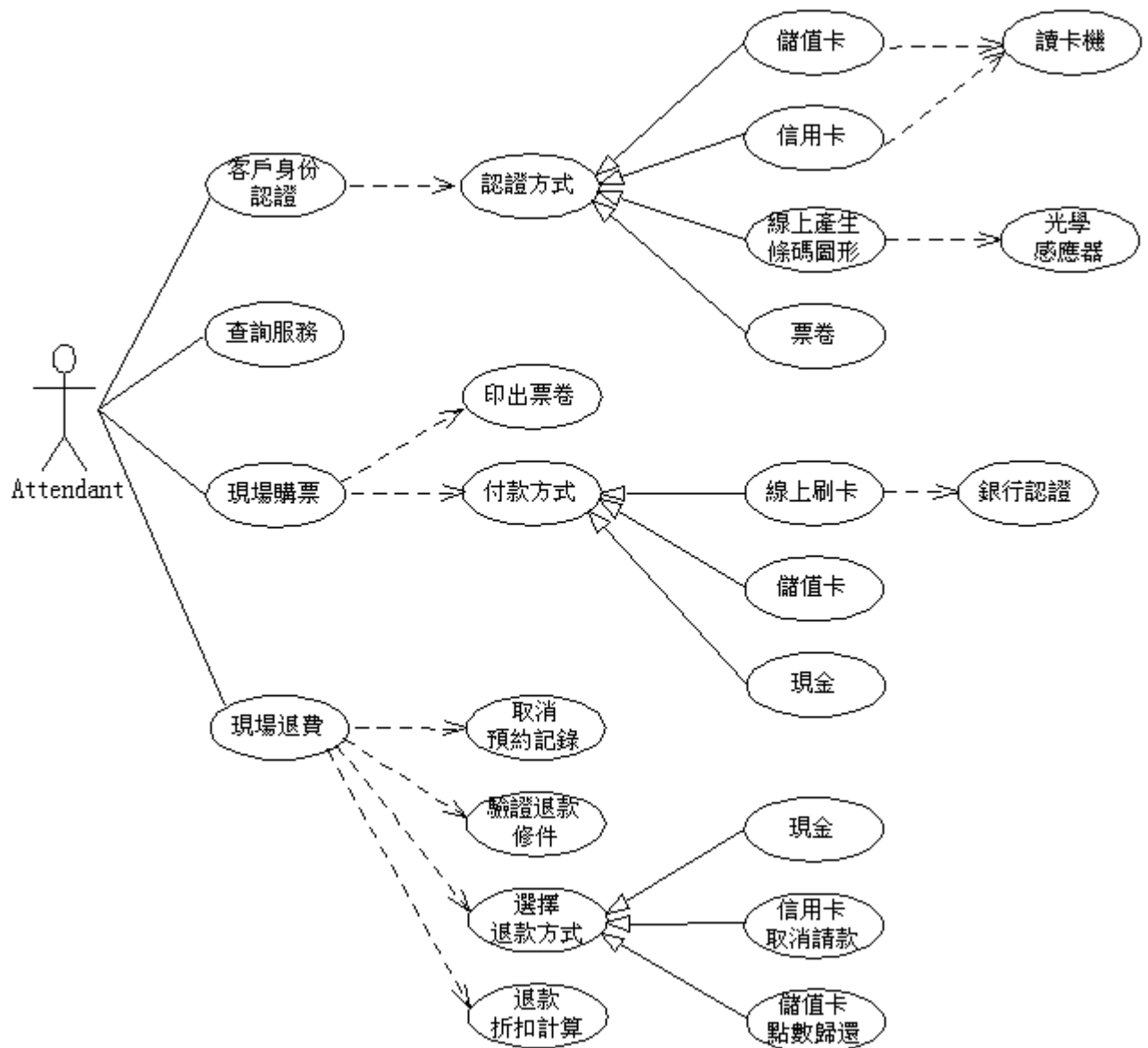
## 3 系統分析

### 3.1 使用案例

#### 3.1.1 Merchant系統管理者

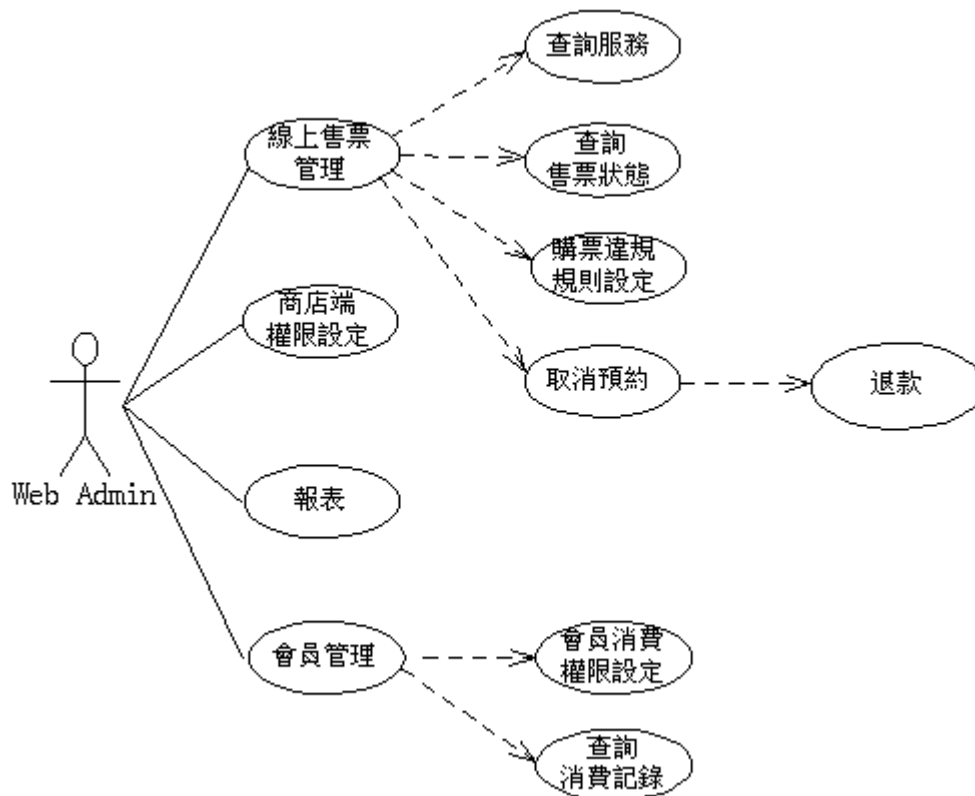


## 3.1.2 商店服務員

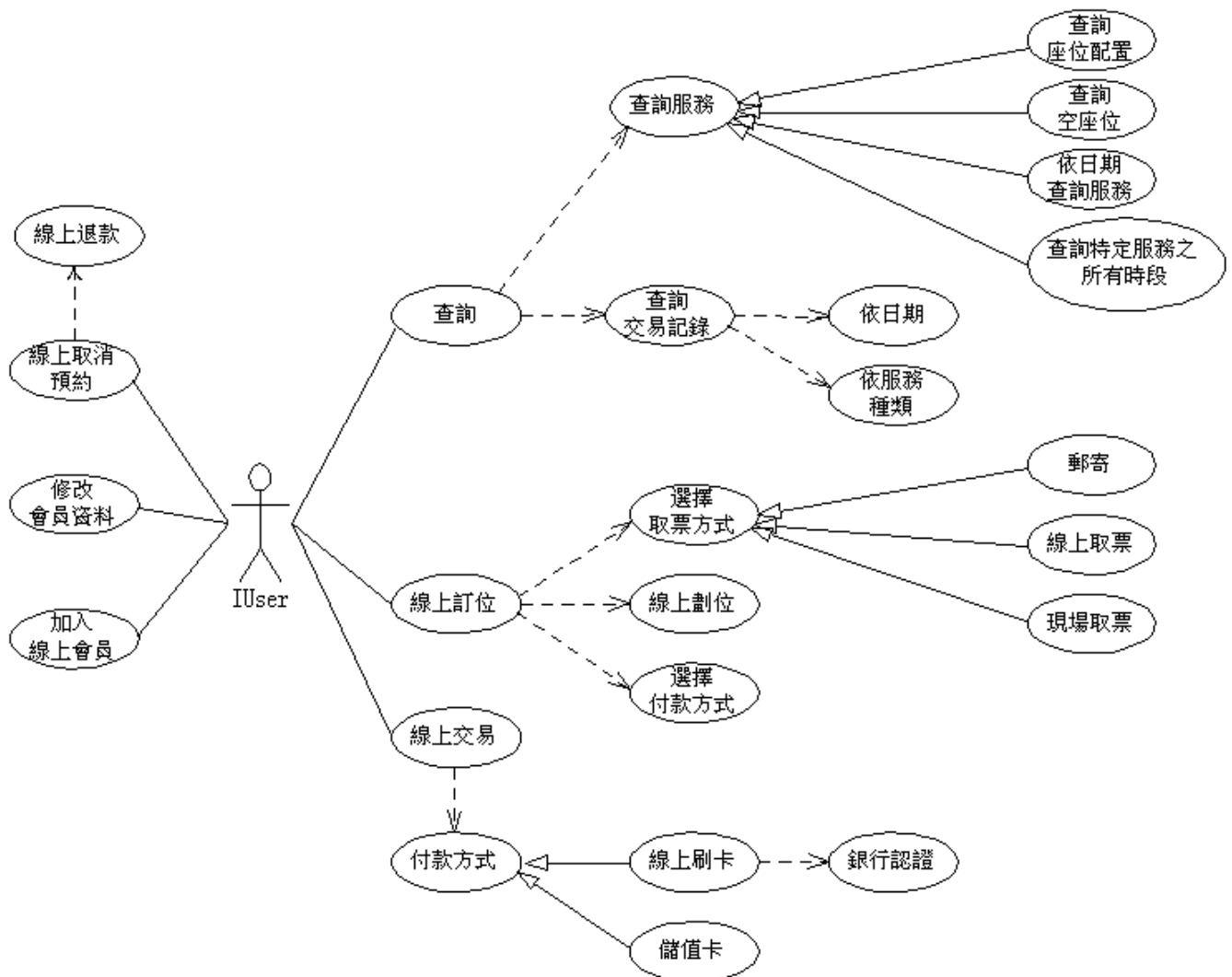




### 3.1.3 Web Plus系統管理者



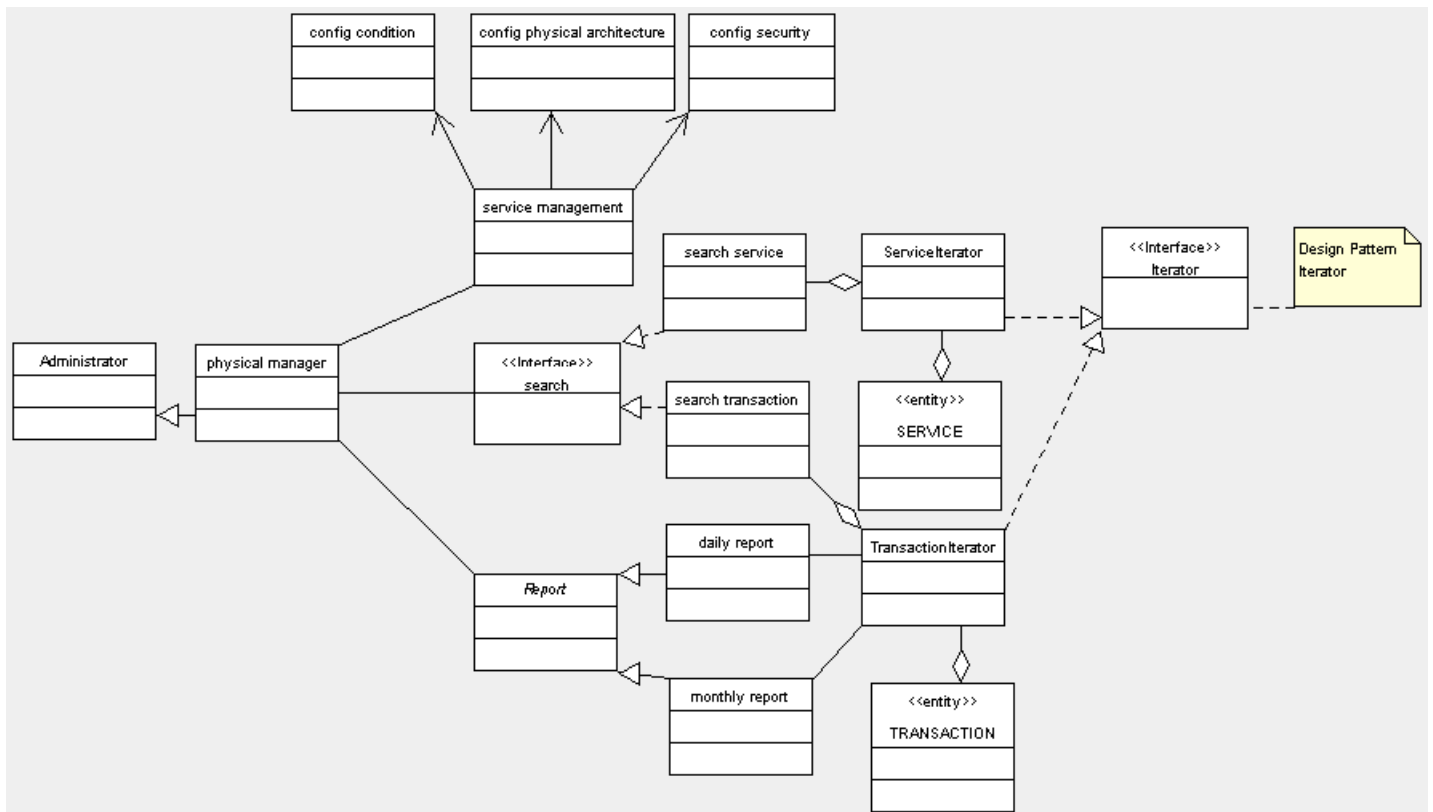
## 3.1.4 線上使用者



## 3.2 類別關係圖

### 3.2.1 票卷服務設定

系統管理者進行票卷服務設定時所引發的相關類別合作關係。

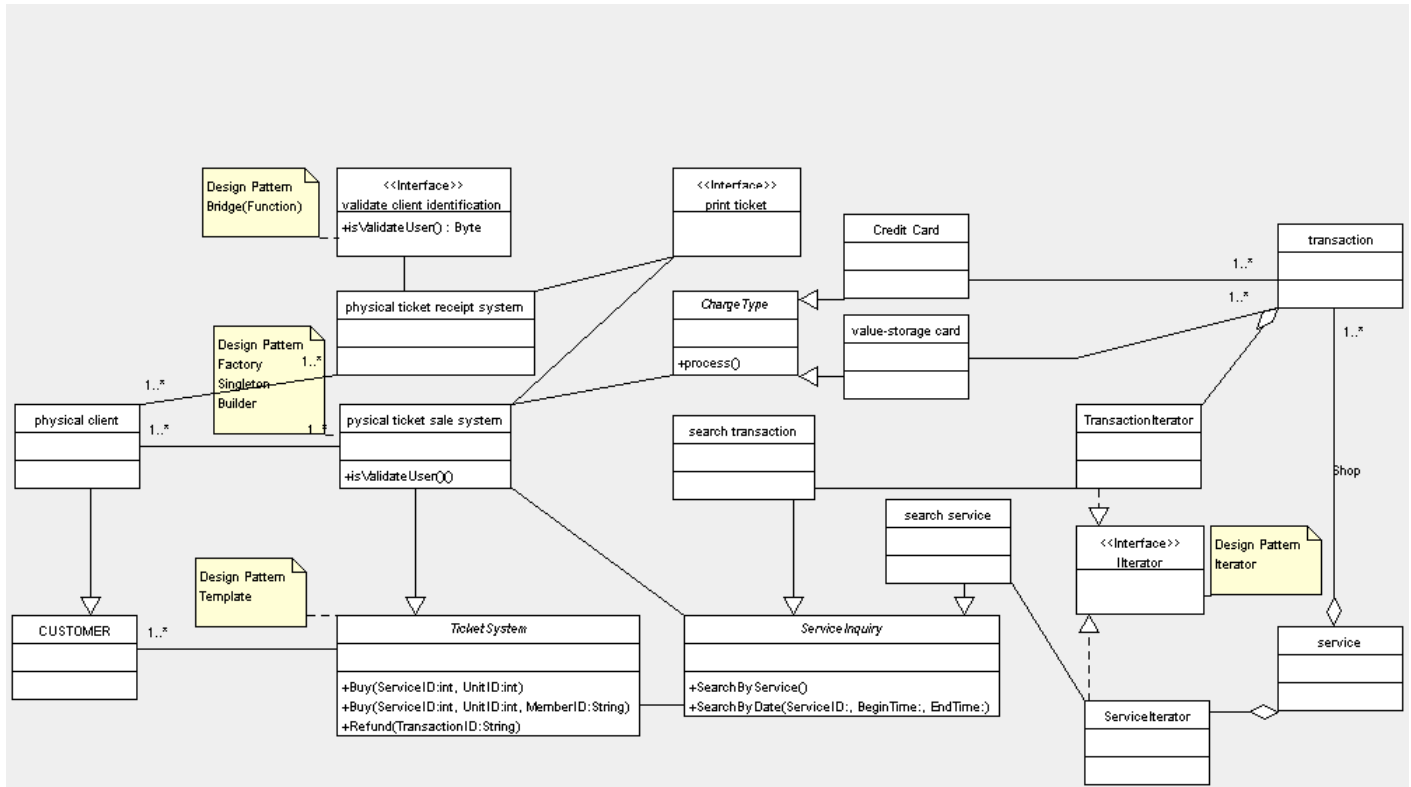


包含線上系統管理者執行「服務查詢」、「報表產生」、「會員管理」功能所引發的類別合作關係。



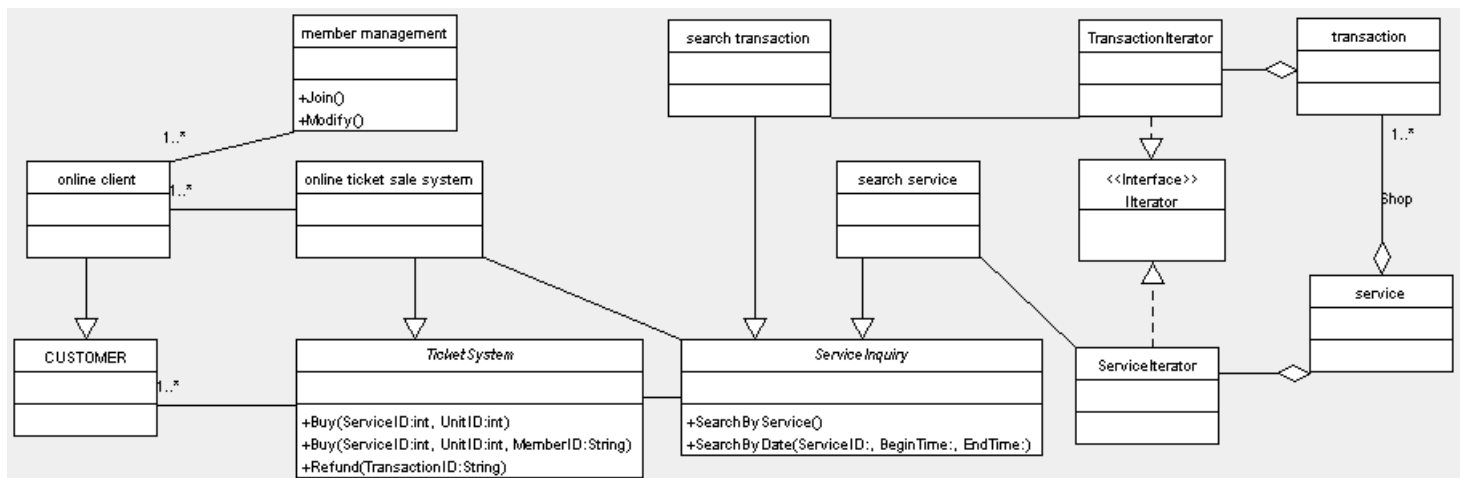
### 3.2.3 商店購票

客戶在Merchant進行購票行為時所引發的類別合作關係。



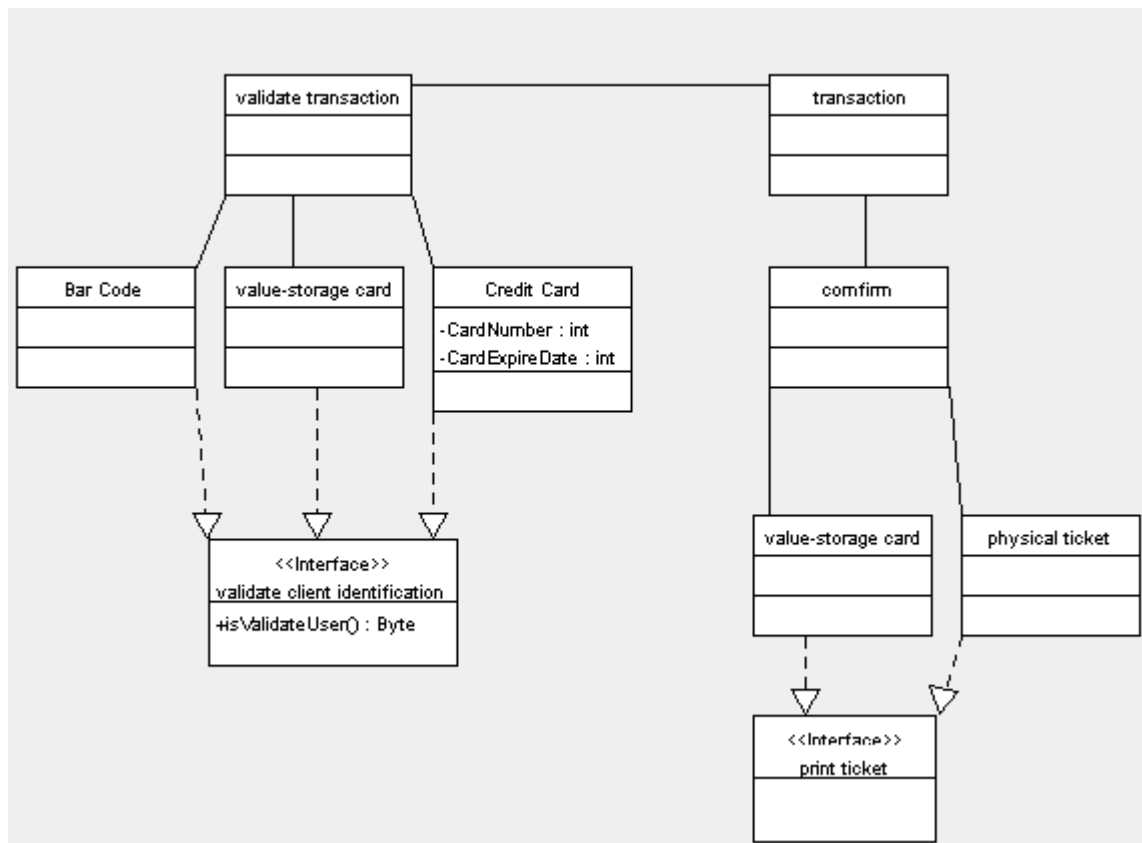
### 3.2.4 線上購票

客戶在Web進行購票行為時所引發的類別合作關係。



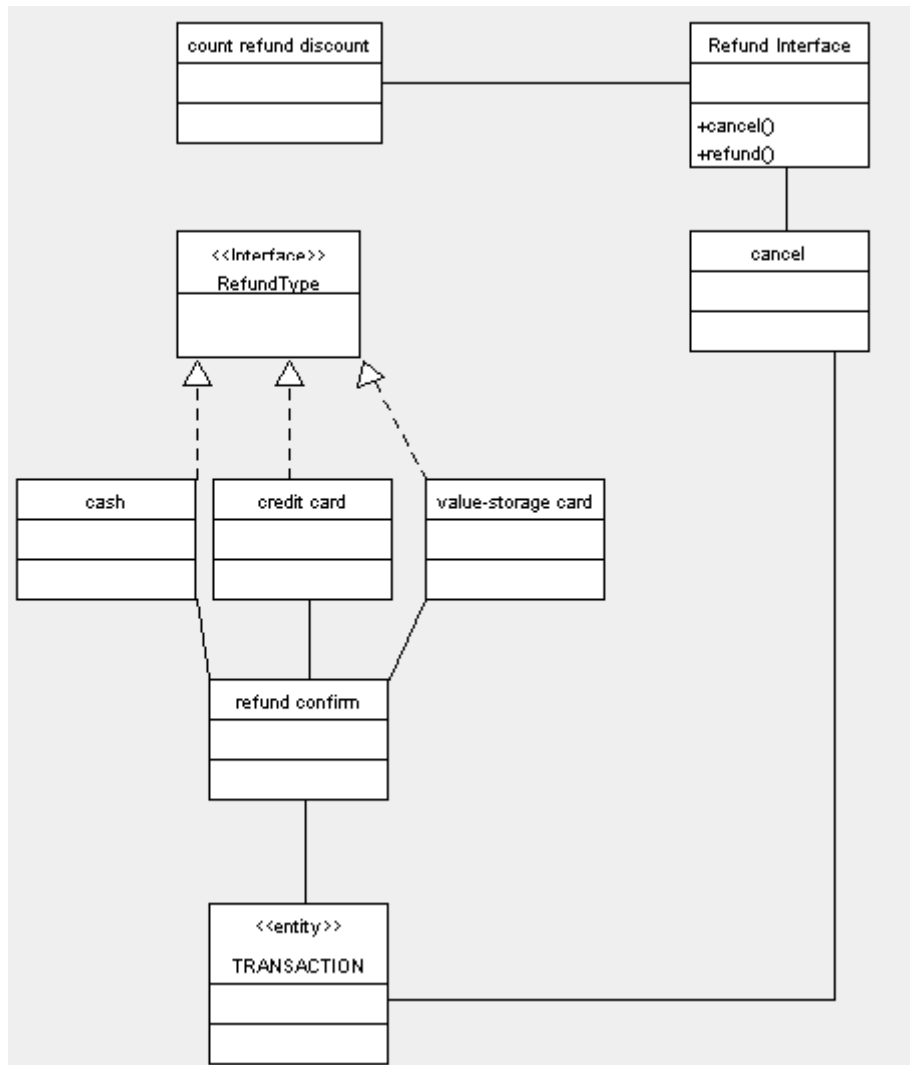
### 3.2.5 身份驗證及票卷印出

客戶在商店購票後或線上會員到現場取票時系統支援身份認證及後端票卷印出的類別合作關係。



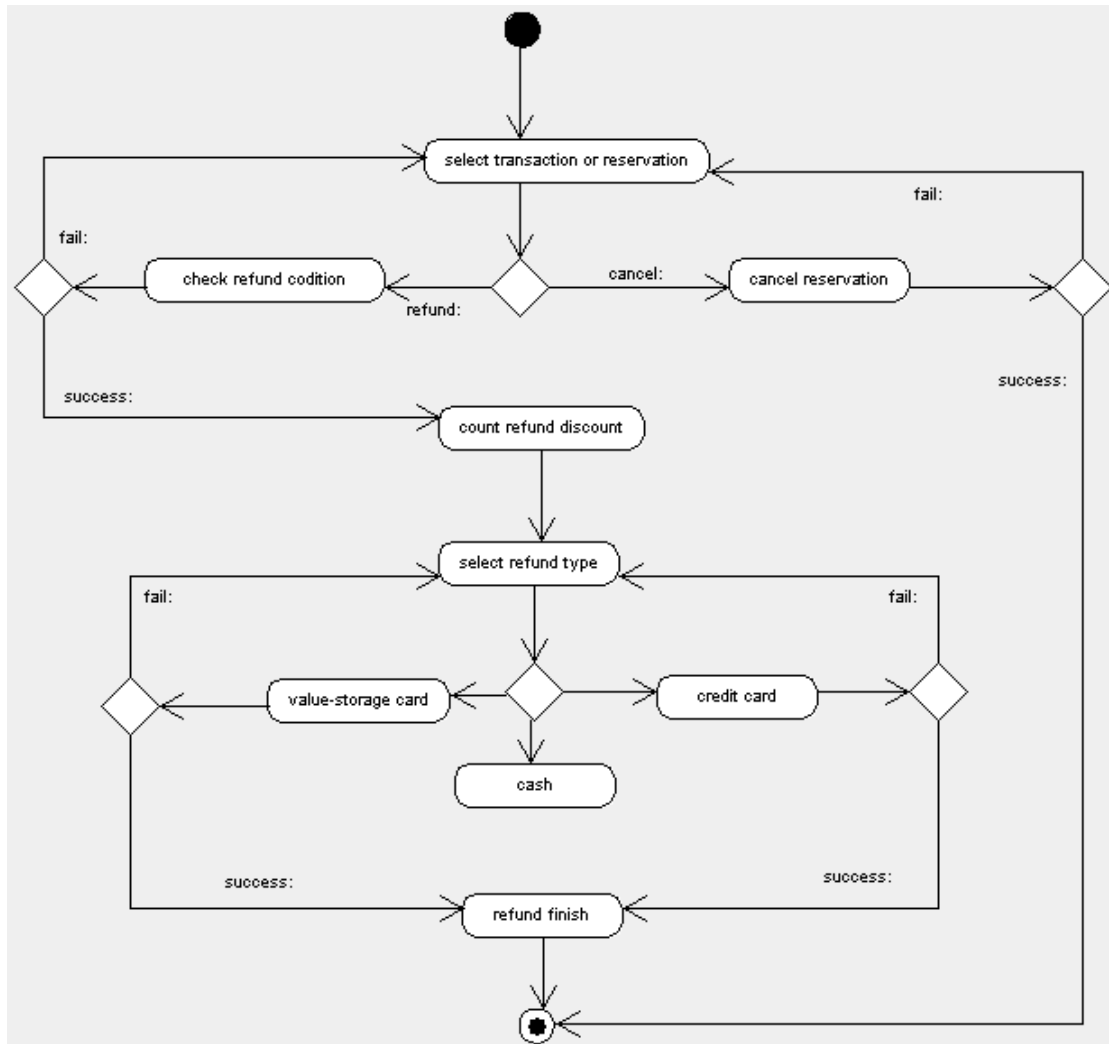
### 3.2.6 退款

現場客戶或線上會員要求退款時所引發的類別合作關係。



### 3.3 活動圖

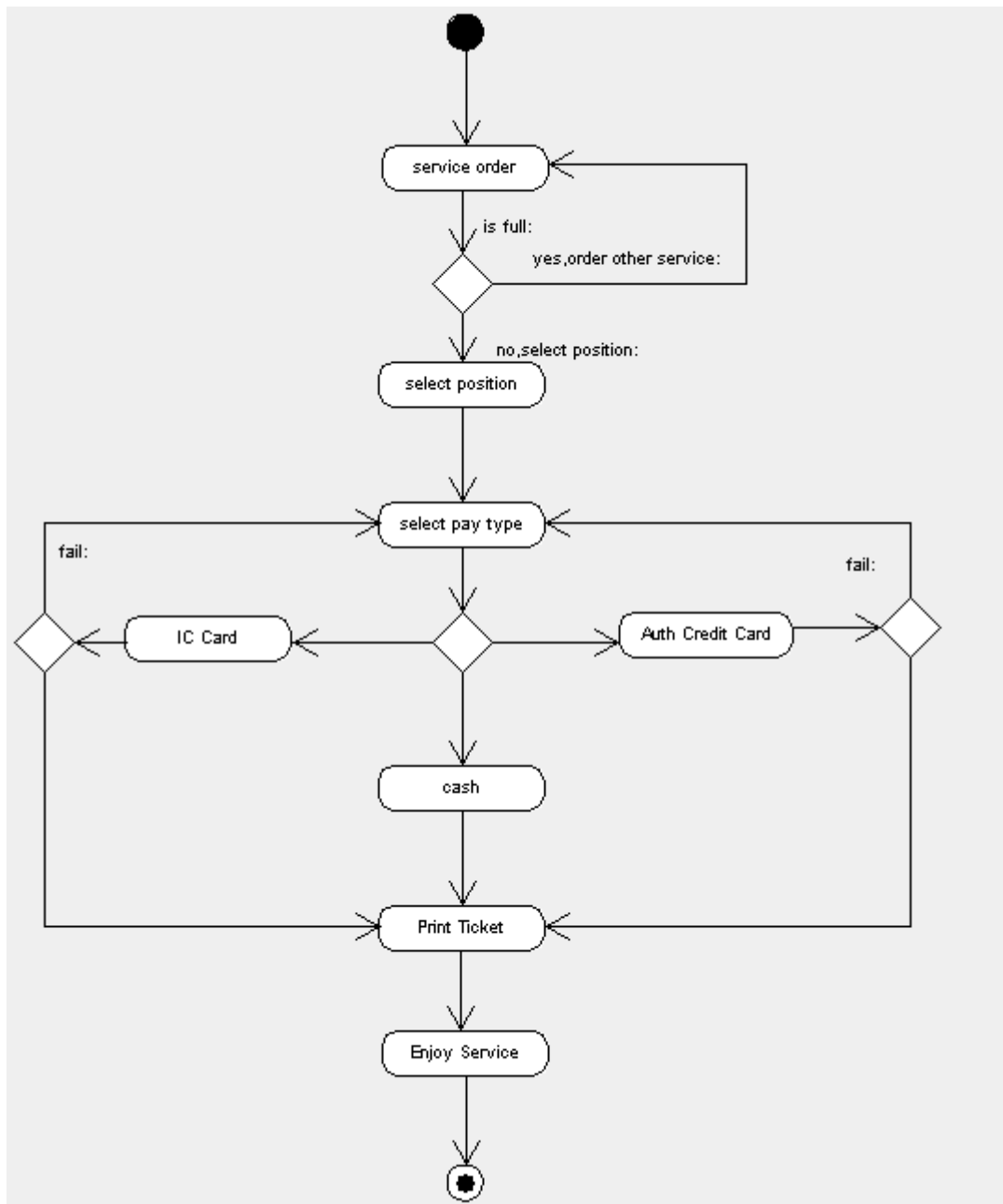
#### 3.3.1 服務設定



1. 系統管理者開始實體資源設定，內含所有單位資源清單
2. 產生Architecture後開始產生Resource
3. 新增Resource實際銷售資源Service，以及做為服務搜尋依據的關鍵字定義
4. 為Service定義提供服務的時間表及計費標準
5. 設定瀏覽、訂購、管理服務的使用者權限，以及開放線上訂購的服務範圍
6. 設定服務的退款規則
7. 視需要設定套裝服務
8. 完成設定後即可開放服務供商店或線上客戶購買

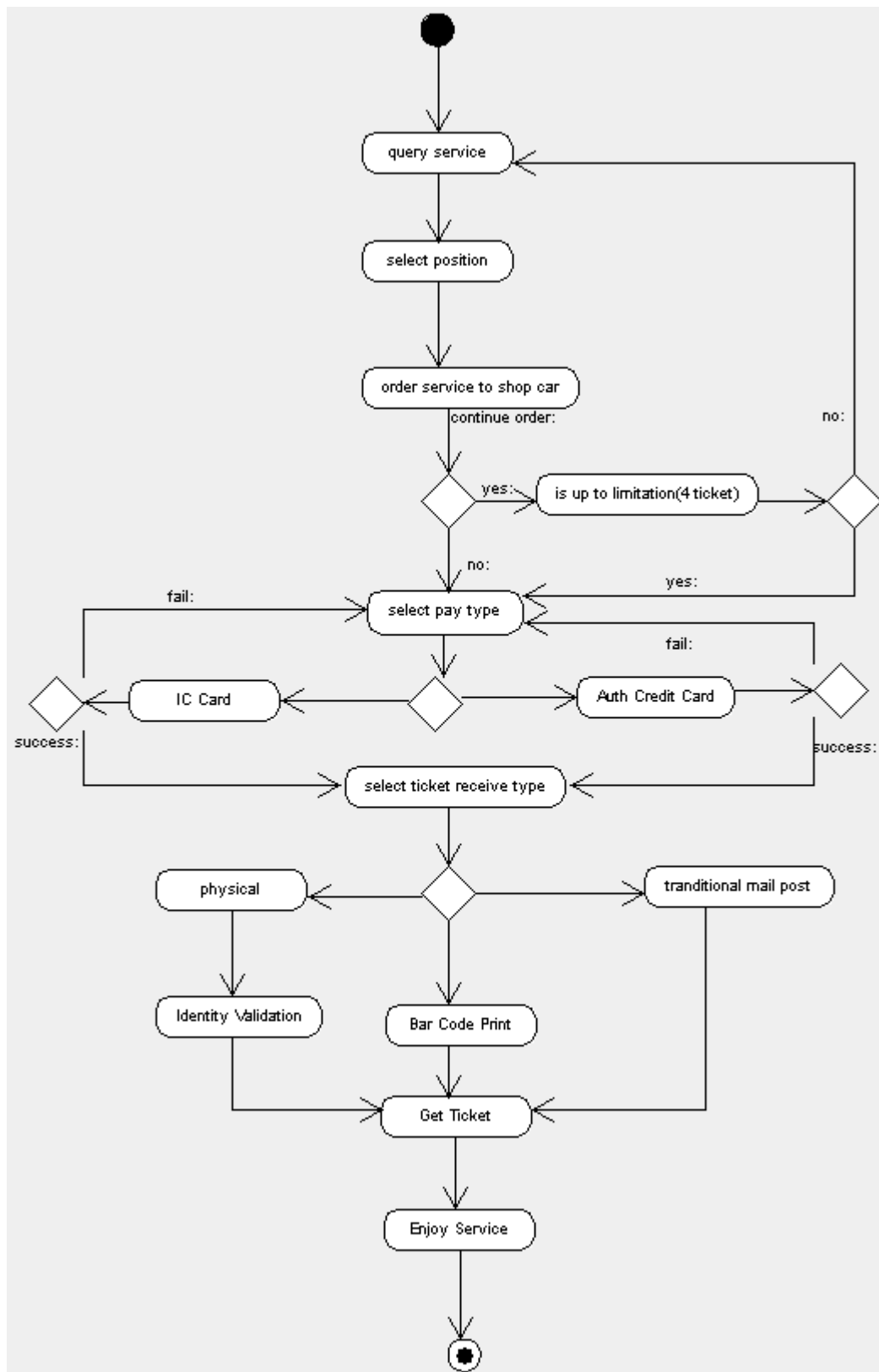


### 3.3.2 商店購票及取票



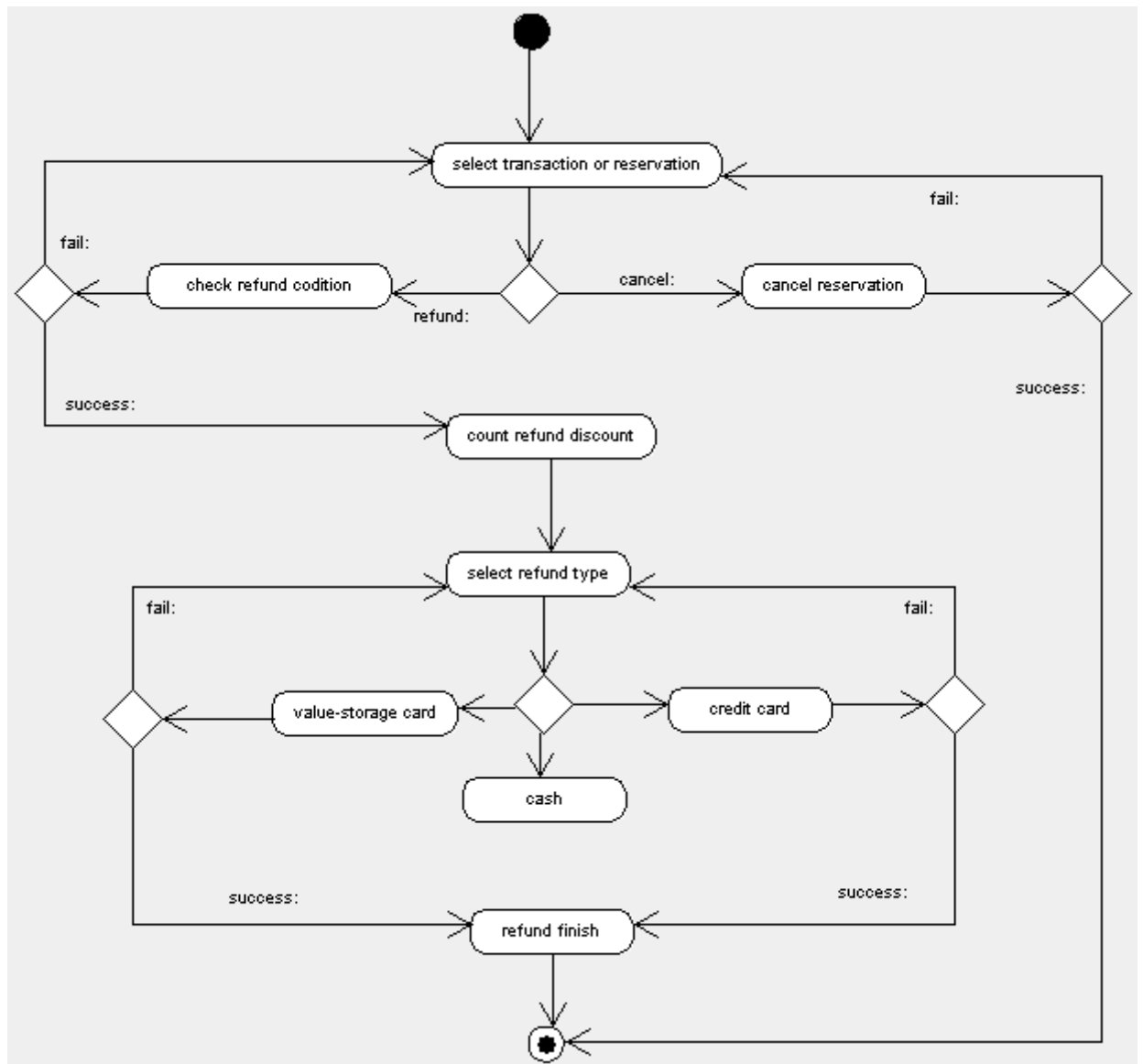
1. 客戶到商店指定服務進行購票
2. 如所有票卷資源皆已售出，則重新選擇服務
3. 如票卷資源尚未售完則進行劃位
4. 客戶選擇商店支援的付款方式，可以是現金、信用卡、儲值IC卡
5. 商店印出票卷交給客戶
6. 客戶完成購票並使用服務

## 3.3.3 線上購票及取票



1. 線上會員查詢可購買的服務
2. 自行選擇可購買的座位放到個人購物籃中
3. 如購物籃物品已超過4個則強迫結帳，否則可繼續購買
4. 客戶選擇商店支援的付款方式，可以是信用卡、儲值IC卡
5. 客戶選擇商店支援的[票卷種類](#)及[取票方式](#)，可以是郵寄票卷、現場取票、線上取票、無票卷
6. 如會員選擇現場取票或以線上票卷換票，則Merchant端會執行會員身份驗證工作
7. 客戶取得正式票卷後即可使用服務

## 3.3.4 退款

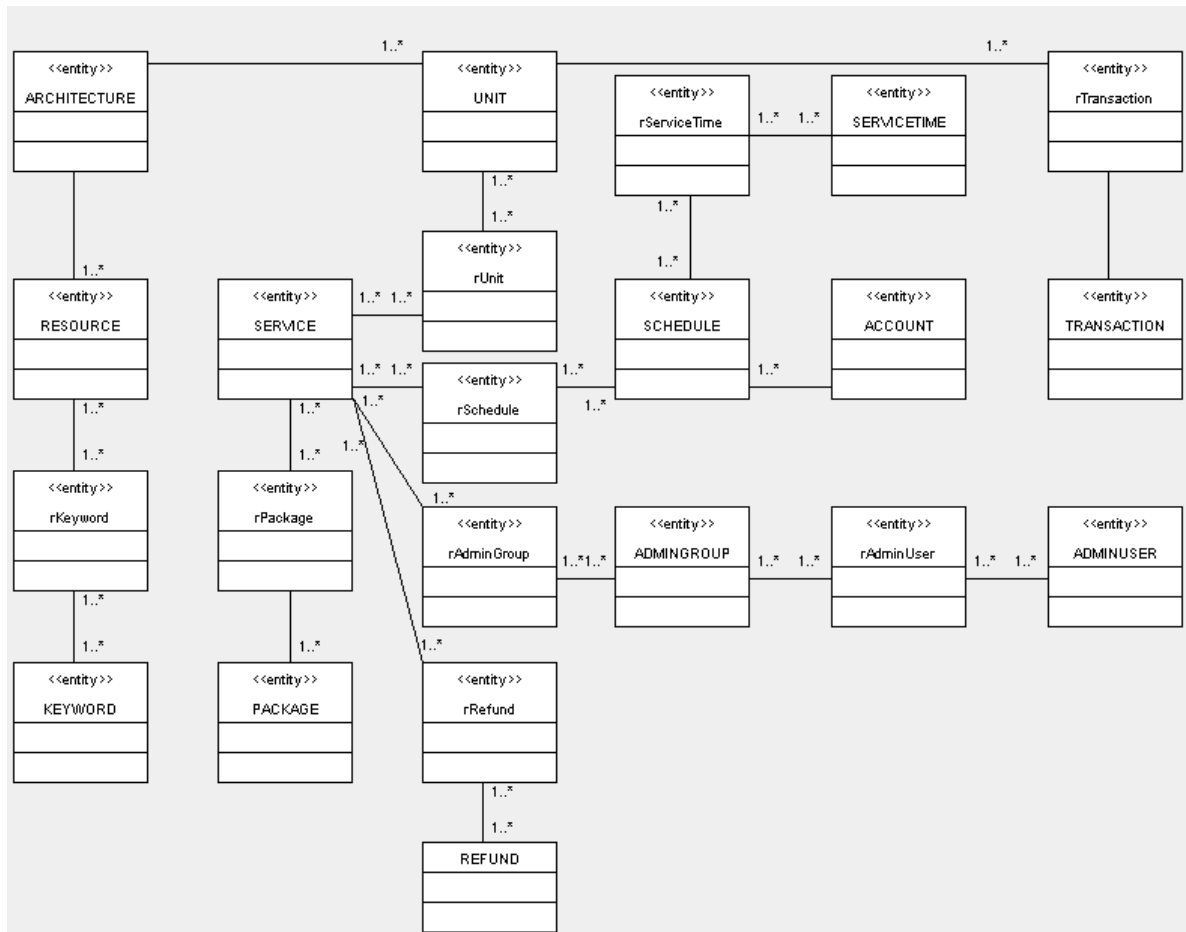


1. 接受客戶退款要求
2. 客戶如透過線上購票則先查詢交易記錄
3. 如已付費則進行退款動作，如未付費則取消交易及購買記錄
4. 進行退款前先檢查退款條件是否成立
5. 取消購買記錄並依退款設定進行折扣計算
6. 依客戶付款途徑決定退款途徑，可能是現金、信用卡、儲值IC卡
7. 完成退款作業

## 4 資料分析

### 4.1 資料庫

#### 4.1.1 Merchant系統資料庫



- **Architecture**

記錄所有實體資源配置的XML資訊

- **Unit**

記錄Architecture中設定的所有單位資源資訊

- **Service**

- 為單位資源 (Unit) 的群組資訊，可視為區域型資源
- 可彈性設定每項區域型資源提供服務的期間

- **Resource**

- 繼承Architecture所設定實體配置架構的票卷銷售資

訊

- 一項Architecture可依時間或空間切割為多個Resource
- 單項Resource需設定提供服務時期，如以「季」為時間單位；只有已設定服務時間的Resource才是可銷售的票卷服務
- 單項Resource的實際銷售資源由多個服務提供時間與空間設定相同的Service提供

- **Keyword**

描述Resource的關鍵字索引

- **Schedule**

- 記錄每項Service實際可循環提供服務的帶狀時間表
- 每項Schedule控制一種計費方式
- 單項Service可依時間或計費方式的不同切割成多個Schedule

- **ServiceTime**

記錄Schedule實際時間格式

- **Package**

- 記錄可銷售套裝票卷服務資訊
- 由多個Service組合而成的套裝票卷服務
- 由於Service資源具有時間及空間設定，可依管理者的商業考量組合成具競爭性的銷售服務

- **Transaction**

記錄實際交易記錄

- **Account**

設費設定

- **Refund**

- 記錄退款規則
- 每項Service皆可引用已設定的退款規則

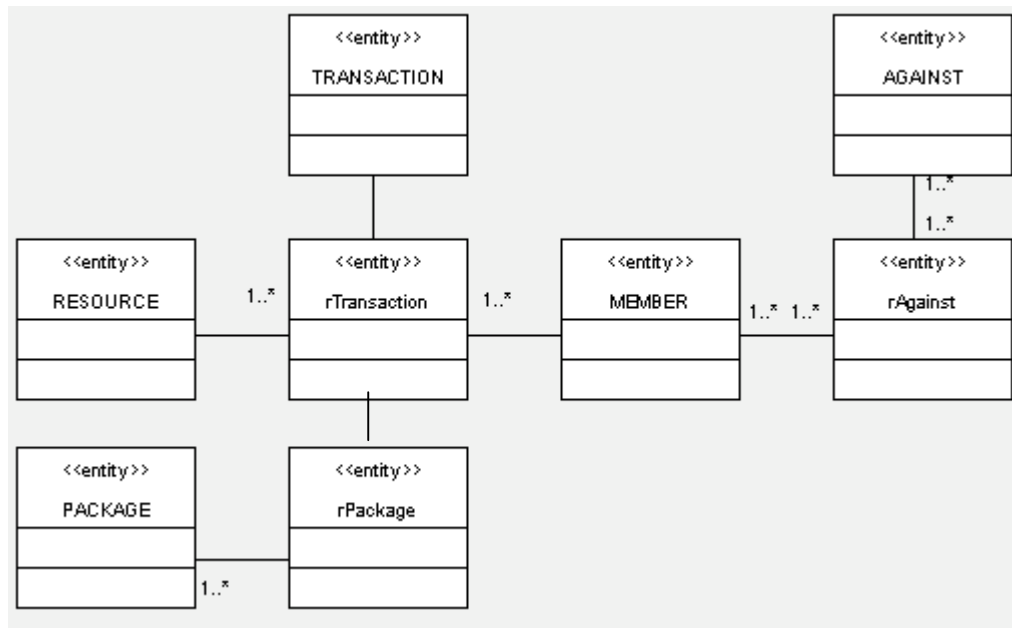
- **AdminUser**

系統管理者帳號

- **AdminGroup**

系統管理者群組角色設定

### 4.1.2 Web Plus線上交易資料庫



#### ● Member

記錄所有線上會員資料

#### ● Against

- 記錄線上購票違規規則
- 系統自動為會員及其曾觸犯的規則建立關聯「rAgainst」，當會員會違規達到系統管理者所設定的標準則會執行會員停權動作

#### ● Resource

- 記錄由不同的Merchant端所提供的所有可銷售票卷服務資訊
- 在Web端儲放Resource資訊可增加線上使用者瀏覽服務時的效能

#### ● Package

- 記錄由不同的Merchant端所提供的可銷售套裝票卷服務資訊
- 在Web端儲放Package資訊可增加線上使用者瀏覽服務時的效能
- 其關聯資料表「rPackage」則存放Package內含的service資訊

#### ● Transaction

- 記錄線上會員的所有交易記錄
- 為Merchant端務帳資料來源之一

## 4.2 XML訊息

### 4.2.1 Architecture

此XML記錄每一項實體資源的配置設定。(詳細Schema設計請參考[附件B.1](#))



```

<?xml version="1.0" encoding="Big5" ?>
- <jxTicket xmlns="http://www.jxTicket.com.tw/ARCHITECTURE/">
- <root id="a0001">
- <basic>
  <!-- sys defind -->
  <cname>喜滿客</cname>
  <ename>CINAMARK</ename>
  <area>台北市</area>
  <telephone>02-12345678</telephone>
  <address>台北市松山區</address>
  <description>The Best Movie Theater In Taiwan</description>
- <defind>
  <keyword name="設備">Dobe</keyword>
</defind>
</basic>
- <connector>
  <branch id="a1001" />
</connector>
</root>
- <branch id="a1001">
- <basic>
  <cname>第1廳</cname>
  <ename>ONE</ename>
  <description>第1廳</description>
</basic>
- <collection>
  <branch id="a1011" />
  <branch id="a1021" />
</collection>
- <deployment>
  <image path="" />
  - <matrix width="10" height="10">
    <image id="0002" width="" height="" top="2" left="3" text="" src="" title="" />
  </matrix>
</deployment>
</branch>
- <branch id="a1011">
- <basic>
  <cname>第1排</cname>
  <ename>ONE</ename>
  <description>第1排</description>
</basic>
- <collection>
  <leaf id="u0101" name="A01" />
  <leaf id="u0102" name="A02" />
  <leaf id="u0103" name="A02" />
  <leaf id="u0104" name="A04" />
  <leaf id="u0105" name="A05" />
</collection>
</branch>
- <branch id="a1021">
- <basic>
  <cname>第2排</cname>
  <ename>TWO</ename>
  <description>第2排</description>
</basic>
- <collection>
  <leaf id="u0201" name="B01" />
  <leaf id="u0202" name="B02" />
  <leaf id="u0203" name="B02" />
  <leaf id="u0204" name="B04" />
  <leaf id="u0205" name="B05" />
</collection>
</branch>
</jxTicket>

```

## ● Root

每一項Architecture只能有一組Root，負責實體資源的空間描述

## ● Basic

描述實體資源的基本資訊，其下Element規格固定

- **Defind**

其下Element規格可由系統管理者依實際所需自由產生

- **Connector**

描述Root下真正的實體資源

- **Branch**

描述區塊資源，該區塊資源的範圍由操作者自行定義

- **Collection**

描述Branch下所包含的資源；每項Branch下可層定義多個Branch，或者是最小的單位資源Leaf

- **Deployment**

描述資源配置圖形

- **Leaf**

描述最小的單位資源

### 4.2.2 Resource

此XML是依據Architecture所產生的票卷銷售服務資訊。(詳細Schema設計請參考[附件B.2](#))

```
<?xml version="1.0" encoding="Big5" ?>
- <jxTicket xmlns="http://www.jxTicket.com.tw/RESOURCE/">
- <resource id="r0001">
- <basic>
  <cname />
  <ename />
  <area />
  <city />
  <address />
  <telephone />
  <status />
  <BeginTime />
  <EndTime />
  <BeginShopHours />
  <EndShopHours />
  <target />
  <description />
  <defind />
</basic>
- <keyword>
  <text value="戲院" />
  <text value="電影" />
  <text value="VIP" />
</keyword>
- <collection>
  <service id="s0001" />
  <service id="s0002" />
  <service id="s0003" />
</collection>
</resource>
</jxTicket>
```

- **Resource**

每項Resource代表一項已初始的票卷服務資訊

- **Basic**

描述票卷服務的基本資訊，其下Element規格固定

- **Defind**

其下Element規格可由系統管理者依實際所需自由產生

- **Keyword**

描述服務的關鍵字

- **Collection**

描述Resource下所包含的Service區域型資源

### 4.2.3 Package

此XML為可銷售套裝服務資訊。(詳細Schema格式及請參考[附件B.3](#))

```
<?xml version="1.0" encoding="Big5" ?>
- <jxTicket xmlns="http://www.jxTicket.com.tw/PACKAGE/">
- <package id="p0001">
- <basic>
  <cname>週票</cname>
  <ename>weekly ticket</ename>
  <description>you can use the ticket....</description>
- <defind>
  <keyword name="贊助者">Sprint</keyword>
  </defind>
</basic>
- <collection>
  <service id="s1001" order="1" />
  <service id="s1001" order="2" />
  <service id="s1001" order="3" />
</collection>
</package>
</jxTicket>
```

- **Package**

每項Package代表一項已初始的套裝服務資訊

- **Basic**

描述套裝服務的基本資訊，其下Element規格固定

- **Defind**

其下Element規格可由系統管理者依實際所需自由產生

- **Collection**

描述Package下所包含的Service區域型資源

## A 附錄

### A.1 票卷種類

票卷種類隨商業經營模式而千變萬化，經分類後可分為以下三大類。而在配合外部工具的使用後jxTicket完全可支援以下票卷的產生及驗證。

#### 1. 傳統票卷

- 消費者經由現場購票、現場取票(預約或線上購票)、郵寄(線上購票)取得
- 票卷由實體通路自行印製

#### 2. 線上票卷

消費者完成線上購票後取得購買代碼，依線上售票系統指示到特定網址印出圖形票卷

線上直接製作圖形票卷，實體通路需配合使用票卷驗證工具，或身份證、會員卡等身份認證工具才使用此類別

#### 3. 無票卷

消費者完成線上購買後配合會員制度以身份證、會員卡、IC卡、儲值卡等外部驗證工具做身份識別

建議『會員制或不涉及金錢交易』的服務才使用此類別

### A.2 取票方式

#### 1. 現場取票

由實體通路櫃台直接售出

線上訂票以光學感應器驗證線上票卷兌換實體票卷，或人工驗證「身份證號」「信用卡」方式處理

#### 2. 郵寄：需注意郵件處理時間

#### 3. 線上票卷：以光學感應器驗證線上票卷

## B 附件

### B.1 XML Design - Architecture

- XML Schema

```

<?xml version="1.0" encoding="UTF-8" ?>
= <xs:schema
    targetNamespace="http://www.jxTicket.com.tw/ARCHITECTURE/"
    xmlns:xs="http://www.w3.org/2001/XMLSchema">
- <!-- definition DocumentElement -->
= <xs:element name="jxTicket">
    = <xs:complexType>
        = <xs:sequence>
            <xs:element ref="root" minOccurs="1" maxOccurs="1" />
            <xs:element ref="branch" minOccurs="0" maxOccurs="unbounded" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
- <!-- definition ChildElement -->
= <xs:element name="root">
    = <xs:complexType>
        = <xs:sequence>
            = <xs:element name="basic">
                = <xs:complexType>
                    = <xs:sequence>
                        <xs:element name="cname" type="xs:string"
                            />
                        <xs:element name="ename" type="xs:string"
                            />
                        <xs:element name="area" type="xs:string"
                            />
                        <xs:element name="telephone"
                            type="xs:string" />
                        <xs:element name="address"
                            type="xs:string" />
                        <xs:element name="description" type="xs:string"

```

```

        />
        <xs:element ref="defind" />
    </xs:sequence>
</xs:complexType>
</xs:element>
= <xs:element name="connector">
    = <xs:complexType>
        = <xs:sequence>
            = <xs:element name="branch">
                = <xs:complexType>
                    <xs:attribute name="id" type="xs:string"
                        />
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
- <!-- definition ChildElement -->
= <xs:element name="defind">
    = <xs:complexType>
        = <xs:sequence>
            = <xs:element name="keyword" type="xs:string" minOccurs="0"
                maxOccurs="unbounded">
                = <xs:complexType>
                    <xs:attribute name="name" type="xs:string" />
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
- <!-- definition ChildElement -->
= <xs:element name="branch">
    = <xs:complexType>
        = <xs:sequence>
            = <xs:element name="basic" minOccurs="1"

```

```

maxOccurs="1">
= <xs:complexType>
= <xs:sequence>
  <xs:element name="cname" minOccurs="1" maxOccurs="1" />
  <xs:element name="ename" minOccurs="1" maxOccurs="1" />
  <xs:element name="description" minOccurs="1" maxOccurs="1" />
</xs:sequence>
</xs:complexType>
</xs:element>
= <xs:element name="collection" minOccurs="1" maxOccurs="1">
= <xs:complexType>
= <xs:sequence>
= <xs:element name="branch" minOccurs="1"
maxOccurs="unbounded">
= <xs:complexType>
  <xs:attribute name="id" type="xs:string" />
</xs:complexType>
</xs:element>
= <xs:element name="leaf" minOccurs="1"
maxOccurs="unbounded">
= <xs:complexType>
  <xs:attribute name="id" type="xs:string" />
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
= <xs:element name="deployment" minOccurs="1" maxOccurs="1">
= <xs:complexType>
= <xs:sequence>
= <xs:element name="image">
= <xs:complexType>
  <xs:attribute name="path" type="xs:string" />
</xs:complexType>
</xs:element>
= <xs:element name="matrix" minOccurs="1" maxOccurs="unbounded">
= <xs:complexType>

```



```
<xs:sequence>
  <xs:element name="image" minOccurs="1" maxOccurs="unbounded">
    <xs:complexType>
      <xs:attribute name="id" type="xs:string" />
      <xs:attribute name="width" type="xs:integer" />
      <xs:attribute name="height" type="xs:integer" />
      <xs:attribute name="top" type="xs:integer" />
      <xs:attribute name="left" type="xs:integer" />
      <xs:attribute name="text" type="xs:string" />
      <xs:attribute name="src" type="xs:string" />
      <xs:attribute name="title" type="xs:string" />
    </xs:complexType>
  </xs:element>
</xs:sequence>
<xs:attribute name="width" type="xs:integer" />
<xs:attribute name="height" type="xs:integer" />
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="id" type="xs:string" />
</xs:complexType>
</xs:element>
</xs:sequence>
```

## B.2 XML Design - Resource

### ● XML Schema

```

<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema
  targetNamespace="http://www.jxTicket.com.tw/RESOURCE/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  - <!-- definition DocumentElement -->
  <xs:element name="jxTicket">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="resource" minOccurs="1" maxOccurs="1" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  - <!-- definition ChildElement -->
  <xs:element name="resource">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="basic">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="cname" type="xs:string" />
              <xs:element name="ename" type="xs:string" />
              <xs:element name="area" type="xs:string" />
              <xs:element name="city" type="xs:string" />
              <xs:element name="address" type="xs:string" />
              <xs:element name="telephone" type="xs:string" />
              <xs:element name="status" type="xs:string" />
              <xs:element name="BeginTime" type="xs:string" />
              <xs:element name="EndTime" type="xs:string" />
              <xs:element name="BeginShopHours" type="xs:string" />
              <xs:element name="EndShopHours" type="xs:string" />
              <xs:element name="target" type="xs:string" />
              <xs:element name="description" type="xs:string" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```

```

        <xs:element ref="defin" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  = <xs:element name="connector">
    = <xs:complexType>
      = <xs:sequence>
        = <xs:element name="service">
          = <xs:complexType>
            <xs:attribute name="id" type="xs:string" />
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
- <!-- definition ChildElement -->
= <xs:element name="defin">
  = <xs:complexType>
    = <xs:sequence>
      = <xs:element name="keyword" type="xs:string" minOccurs="0"
        maxOccurs="unbounded">
        = <xs:complexType>
          <xs:attribute name="name" type="xs:string" />
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>

```

## B.3 XML Design - Package

### ● XML Schema

```

<?xml version="1.0" encoding="UTF-8" ?>
= <xs:schema targetNamespace="http://www.jxTicket.com.tw/PACKAGE/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
- <!-- definition DocumentElement -->
= <xs:element name="jxTicket">
  = <xs:complexType>
    = <xs:sequence>
      <xs:element ref="package" minOccurs="1" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
- <!-- definition ChildElement -->
= <xs:element name="package">
  = <xs:complexType>
    = <xs:sequence>
      = <xs:element name="basic">
        = <xs:complexType>
          = <xs:sequence>
            <xs:element name="cname" type="xs:string" />
            <xs:element name="ename" type="xs:string" />
            <xs:element name="description" type="xs:string" />
            <xs:element ref="defin" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    = <xs:element name="connector">
      = <xs:complexType>
        = <xs:sequence>
          = <xs:element name="service">
            = <xs:complexType>
              <xs:attribute name="id" type="xs:string" />
            </xs:complexType>
          </xs:element>
        </xs:sequence>

```

```
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
- <!-- definition ChildElement -->
= <xs:element name="defind">
  = <xs:complexType>
    = <xs:sequence>
      = <xs:element name="keyword" type="xs:string" minOccurs="0"
        maxOccurs="unbounded">
        = <xs:complexType>
          <xs:attribute name="name" type="xs:string" />
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```

## C DB Schema說明

### C.1 Merchant

#### [ARCHITECTURE]

\*\*\*\*\*

SN	流水號
CreateTime	資料建立時間
LastModifyTime	資料最後修改時間
Name	資料名稱
ArchitectureBody	XML格式，記錄實體資源呈現結構
Description	資料簡單描述

#### [RESOURCE]

\*\*\*\*\*

SN	流水號
CreateTime	資料建立時間
LastModifyTime	資料最後修改時間
CName	中文名稱
ENAME	英文名稱
City	資源在處之城市名稱
Area	資源在處之地區名稱
Telephone	資源在處之電話號碼
Address	資源在處之詳細地址
Status	服務的狀態
	active 開放中
	close 已關閉
Target	提供服務的場合
	merchant 商店端
	webplus 網路端
BeginTime	開始提供服務時間
EndTime	結束終止服務時間
BeginShopHours	服務開始營業時間
EndShopHours	服務終止營業時間
refArchitectureID	參照對應特定Architecture記錄
ResourceBody	XML格式，記錄實際可販售之資源之呈現結構
Description	資料簡單描述

**[SERVICE]**

\*\*\*\*\*

SN	流水號
CreateTime	資料建立時間
LastModifyTime	資料最後修改時間
CName	中文名稱
EName	英文名稱
Status	服務的狀態
	active 開放中
	close 已關閉
Target	提供服務的場合
	merchant 商店端
	webplus 網路端
BeginTime	開始提供服務時間
EndTime	結束終止服務時間
BeginShopHours	服務開始營業時間
EndShopHours	服務終止營業時間
refResourceID	參照對應特定Resource記錄
ServiceBody	XML格式，記錄實際可販售之資源集合
Description	資料簡單描述

**[UNIT]**

\*\*\*\*\*

SN	流水號
CreateTime	資料建立時間
LastModifyTime	資料最後修改時間
Name	名稱
refResourceID	參照對應特定Architecture記錄
Description	資料簡單描述

**[rUnit]**

\*\*\*\*\*

UnitID	關連UNIT Table之記錄
ServiceID	關連SERVICE Table之記錄

**[UNIT]**

\*\*\*\*\*

SN	流水號
CreateTime	資料建立時間
LastModifyTime	資料最後修改時間
Name	名稱
refResourceID	參照對應特定Architecture記錄
Description	資料簡單描述

**[Keyword]**

\*\*\*\*\*

SN	流水號
CreateTime	資料建立時間
LastModifyTime	資料最後修改時間
Name	關鍵字名稱
Value	關鍵字值

**[rKeyword]**

\*\*\*\*\*

KeywordID	關連Keyword Table之記錄
ResourceID	關連RESOURCE Table之記錄

**[PACKAGE]**

\*\*\*\*\*

SN	流水號
CreateTime	資料建立時間
LastModifyTime	資料最後修改時間
Name	套裝服務名稱
BeginTime	開始提供服務時間
EndTime	結束終止服務時間
Description	資料簡單描述

**[rPackage]**

\*\*\*\*\*

PackageID	關連PACKAGE Table之記錄
ServiceID	關連SERVICE Table之記錄



**[Refund]**

\*\*\*\*\*

SN	流水號
CreateTime	資料建立時間
LastModifyTime	資料最後修改時間
Name	名稱
Discount	折扣，內容表示百分比
Condition	大於、小於、等於
Region	時間格式 (month、day、hour、minute、week)
TimePart	時間值
Description	資料簡單描述

**[rRefund]**

\*\*\*\*\*

RefundID	關連REFUND Table之記錄
ServiceID	關連SERVICE Table之記錄

**[SCHEDULE]**

\*\*\*\*\*

SN	流水號
CreateTime	資料建立時間
LastModifyTime	資料最後修改時間
Name	名稱
BeginTime	開始提供服務時間
EndTime	結束終止服務時間
Priority	優先性
refServiceTime	參照對應特定ServiceTime記錄
refAccountID	參照對應特定Account記錄
Description	資料簡單描述

**[rSchedule]**

\*\*\*\*\*

ScheduleID	關連SCHEDULE Table之記錄
ServiceID	關連SERVICE Table之記錄

**[rServiceTime]**

\*\*\*\*\*

ScheduleID	關連SCHEDULE Table之記錄
ServiceTimeID	關連ServiceTime Table之記錄

**[ServiceTime]**

\*\*\*\*\*

SN	流水號
CreateTime	資料建立時間
Region	時間格式 (month、day、hour、minute、week)
TimePart	時間值

**[ACCOUNT]**

\*\*\*\*\*

SN	流水號
CreateTime	資料建立時間
LastModifyTime	資料最後修改時間
Name	名稱
Category	種類
Price	價格
Description	資料簡單描述

**[TRANSACTIONS]**

\*\*\*\*\*

SN	流水號
CreateTime	資料建立時間
LastModifyTime	資料最後修改時間
TransTime	交易時間
TransPrice	交易價格
CaptureTime	結帳時間
CapturePrice	結帳價格
RefundTime	退款時間
RefundPrice	退款價格
RefundDesc	退款說明
refMemberID	參照對應特定會員資料庫記錄
refServiceID	參照對應特定Service記錄
status	交易狀態

---

**[rTransactions]**

\*\*\*\*\*

UnitID                    關連UNIT Table之記錄  
TransactionsID        關連TRANSACTIONS Table之記錄

**[AdminGroup]**

\*\*\*\*\*

SN                        流水號  
CreateTime            資料建立時間  
LastModifyTime       資料最後修改時間  
Name                    名稱  
Rights                  權限設定，以XML格式呈現記錄權限  
Description            資料簡單描述

**[rAdminGroup]**

\*\*\*\*\*

AdminGroupID        關連AdminGroup Table之記錄  
ServiceID            關連SERVICE Table之記錄

**[AdminUser]**

\*\*\*\*\*

SN                        流水號  
CreateTime            資料建立時間  
LastModifyTime       資料最後修改時間  
Name                    名稱  
UserID                  使用者帳號  
UserPWD                使用者密碼  
Description            資料簡單描述

**[rAdminUser]**

\*\*\*\*\*

AdminGroupID        關連AdminGroup Table之記錄  
AdminUserID        關連AdminUser Table之記錄

## C.2 Webplus

### [RESOURCE]

\*\*\*\*\*

SN	流水號
CreateTime	資料建立時間
LastModifyTime	資料最後修改時間
CName	中文名稱
ENAME	英文名稱
City	資源在處之城市名稱
Area	資源在處之地區名稱
Telephone	資源在處之電話號碼
Address	資源在處之詳細地址
Status	服務的狀態
	active 開放中
	close 已關閉
Target	提供服務的場合
	merchant 商店端
	webplus 網路端
BeginTime	開始提供服務時間
EndTime	結束終止服務時間
BeginShopHours	服務開始營業時間
EndShopHours	服務終止營業時間
refArchitectureID	參照對應特定Architecture記錄
ResourceBody	XML格式，記錄實際可販售之資源之呈現結構
Description	資料簡單描述

### [Keyword]

\*\*\*\*\*

SN	流水號
CreateTime	資料建立時間
LastModifyTime	資料最後修改時間
Name	關鍵字名稱
Value	關鍵字值

**[rKeyword]**

\*\*\*\*\*

KeywordID	關連Keyword Table之記錄
ResourceID	關連RESOURCE Table之記錄

**[PACKAGE]**

\*\*\*\*\*

SN	流水號
CreateTime	資料建立時間
LastModifyTime	資料最後修改時間
Name	套裝服務名稱
BeginTime	開始提供服務時間
EndTime	結束終止服務時間
Description	資料簡單描述

**[rPackage]**

\*\*\*\*\*

PackageID	關連PACKAGE Table之記錄
rTransactionID	關連rTransaction Table之記錄

**[TRANSACTION]**

\*\*\*\*\*

SN	流水號
CreateTime	資料建立時間
LastModifyTime	資料最後修改時間
TransTime	交易時間
TransPrice	交易價格
CaptureTime	結帳時間
CapturePrice	結帳價格
RefundTime	退款時間
RefundPrice	退款價格
RefundDesc	退款說明
refMemberID	參照對應特定會員資料庫記錄
refServiceID	參照對應特定Service記錄
status	交易狀態

**[rTransaction]**

\*\*\*\*\*

SN	流水號
ResourceID	關連RESOURCE Table之記錄
TransactionID	關連TRANSACTION Table之記錄
ServiceID	關連SERVICE Table之記錄

**[MEMBER]**

\*\*\*\*\*

SN	流水號
CreateTime	資料建立時間
LastModifyTime	資料最後修改時間
Name	名稱
UserID	使用者帳號
UserPWD	使用者密碼
Profile	會員基本資料，以XML格式記錄
Description	資料簡單描述

**[AGAINST]**

\*\*\*\*\*

SN	流水號
CreateTime	資料建立時間
LastModifyTime	資料最後修改時間
Name	名稱
Rule	規則設定，以XML格式呈現記錄規則
Description	資料簡單描述

**[rAgainst]**

\*\*\*\*\*

MemberID	關連MEMBER Table之記錄
AgainstID	關連AGAINST Table之記錄

## D 系統開發基礎

### D.1 使用技術

J2EE、J2SE、Servlet、JSP、Java Web Start

### D.2 作業環境

- **JSDK**

JDK 1.3.1

J2EE SDK 1.3.1

- **IDE**

Forte for Java Enterprise Edition 3.0 試用版

- **OS**

Linux Redhat 7.2

- **Database**

MySQL 3.23.49

- **Application Server**

J2EE RI

- **Web Server/Container**

Apache HTTP Server 1.3.22

Jakarta-Tomcat 4.0.3

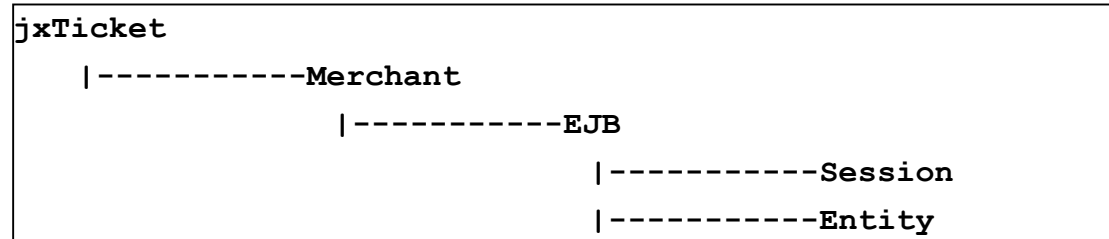
## E 系統相關Package

### E.1 Merchant端

#### E.1-1 DB

主要存放與DB相關之服務程式

##### ● E.1-1-1 簡圖



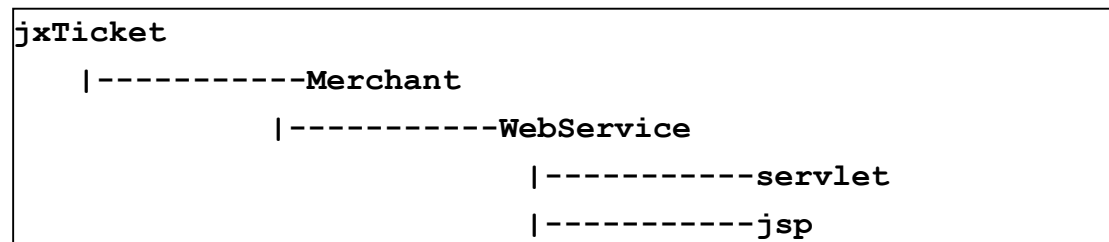
##### ● E.1-1-2 分類說明

- jxTicket.Merchant.EJB.Session
  - 記錄Merchant端資料庫的相關資料，皆為stateless bean
- jxTicket.Merchant.EJB.Entity
  - 記錄Merchant端資料庫的相關資料，每個bean皆對應至DB中之一Table

#### E.1-2 Webservice

主要存放與網路相關之服務程式

##### ● E.1-2-1簡圖



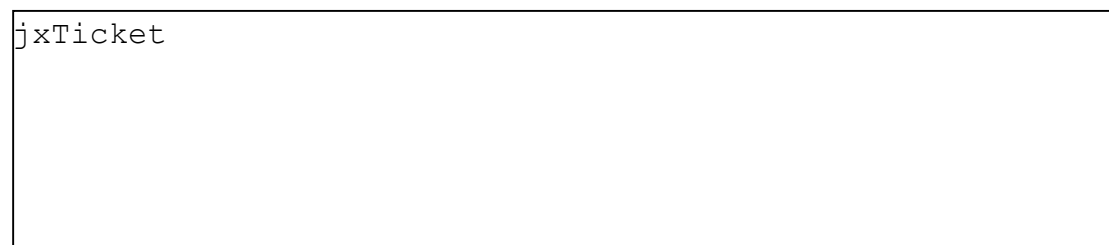
##### ● E.1-2-2 分類說明

- jxTicket.Merchant.WebService.servlet
  - 記錄Merchant端WebService中所有不具UI之網路服務程式
- jxTicket.Merchant.WebService.jsp
  - 記錄Merchant端WebService中所有具備UI之網路服務程式

#### E.1-3 管理介面

主要存放Merchant端client Admin程式

##### ● E.1-3-1 簡圖





```

|-----Merchant
      |-----Admin
            |-----main.class
            |-----resource
            |-----ticket
            |-----payment
            |-----report
            |-----security
            |-----config

```

#### ● E.1-3-2 分類說明

- jxTicket.Merchant.Admin.main
  - 管理介面起始程式
- jxTicket.Merchant.Admin.resource
  - 服務資源管理介面，包含UI、行為以及EJB Client
- jxTicket.Merchant.Admin.ticket
  - 售票管理介面，包含UI、行為以及EJB Client
- jxTicket.Merchant.Admin.payment
  - 帳務管理介面，包含UI、行為以及EJB Client
- jxTicket.Merchant.Admin.report
  - 報表管理介面，包含UI、行為以及EJB Client
- jxTicket.Merchant.Admin.security
  - 系統權限管理介面，包含UI、行為以及EJB Client
- jxTicket.Merchant.Admin.config
  - 系統設定管理介面，包含UI、行為以及EJB Client

### **E.1-4 售票介面**

主要存放Merchant端client Ticket Office程式

#### ● E.1-4-1 簡圖

```

jxTicket
|-----Merchant
      |-----TicketOffice
            |-----main.class
            |-----resource
            |-----ticket
            |-----payment
            |-----report
            |-----config

```

#### ● E.1-4-2 分類說明

- jxTicket.Merchant.TicketOffice.main

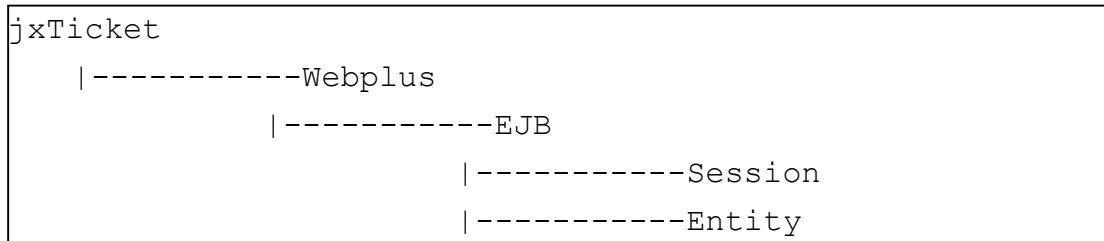
- 管理介面起始程式
- jxTicket.Merchant.TicketOffice.resource
  - 服務資源管理介面，包含UI、行為以及EJB Client
- jxTicket.Merchant.TicketOffice.ticket
  - 售票管理介面，包含UI、行為以及EJB Client
- jxTicket.Merchant.TicketOffice.payment
  - 帳務管理介面，包含UI、行為以及EJB Client
- jxTicket.Merchant.TicketOffice.report
  - 報表管理介面，包含UI、行為以及EJB Client
- jxTicket.Merchant.TicketOffice.config
  - 系統設定管理介面，包含UI、行為以及EJB Client

## E.2 Webplus端

### E.2-1 DB

主要存放與DB相關之服務程式

#### ● E.2-1-1 簡圖



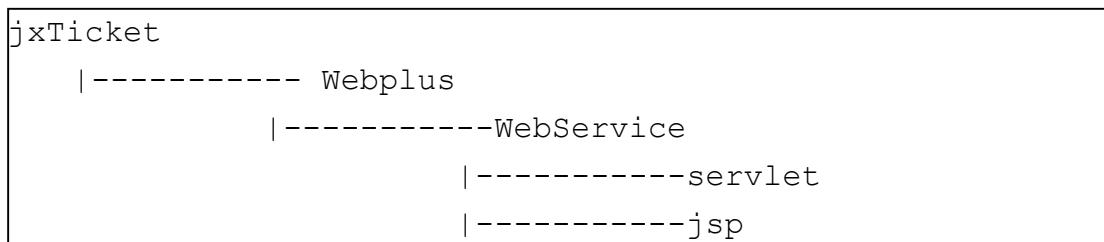
#### ● E.2-1-2 分類說明

- jxTicket.Webplus.EJB.Session
  - 記錄Webplus端資料庫的相關資料，皆為stateless bean
- jxTicket.Webplus.EJB.Entity
  - 記錄Webplus端資料庫的相關資料，每個bean皆對應至DB中之一Table

### E.2-2 WebService

主要存放與網路相關之服務程式

#### ● E.2-2-1 簡圖



#### ● E.2-2-2 分類說明

- jxTicket.Webplus.WebService.servlet

- 記錄Webplus端WebService中所有不具UI之網路服務程式
- jxTicket.Webplus.WebService.jsp
- 記錄Webplus端WebService中所有具備UI之網路服務程式