霹靂卡霹靂拉拉波波莉娜貝貝魯多

系統需求規格書

|  |  |  |
| --- | --- | --- |
| 姓名 | 學號 | E-mail |
| 陳佳吟 | 108820002 | t108820002@ntut.org.tw |
| 林天佑 | 108820003 | t108820003@ntut.org.tw |
| 蔡翔宇 | 108820018 | t108820018@ntut.org.tw |
| 陳力瑋 | 108820038 | t108820038@ntut.org.tw |
| 江大衞 | 108AB0008 | t108AB0008@ntut.org.tw |

Department of Computer Science & Information Engineering

National Taipei University of Technology

內容

[目的 4](#_Toc85978723)

[System Functionality 系統功能 4](#_Toc85978724)

[[User] 系統可以認證使用者 4](#_Toc85978725)

[[Customer] 顧客可以搜尋商品、購買商品、查詢歷史訂單 4](#_Toc85978726)

[[Staff] 工作人員可以管理商品、處理訂單請求 5](#_Toc85978727)

[[Staff] 工作人員可以對訂單或商品訂定折扣方案 5](#_Toc85978728)

[[Staff] 工作人員可以產生系統的報表或統計數據 5](#_Toc85978729)

[Contained Information 系統資訊需求 6](#_Toc85978730)

[使用者 6](#_Toc85978731)

[商品 6](#_Toc85978732)

[商店 7](#_Toc85978733)

[訂單 7](#_Toc85978734)

[銷售報告(sales report) 8](#_Toc85978735)

[系統名稱 8](#_Toc85978736)

[概觀 9](#_Toc85978737)

[系統 9](#_Toc85978738)

[系統描述 9](#_Toc85978739)

[系統架構圖 10](#_Toc85978740)

[操作概念 10](#_Toc85978741)

[Scenario 1： 10](#_Toc85978742)

[Scenario 2： 10](#_Toc85978743)

[Scenario 3： 11](#_Toc85978744)

[Scenario 4： 11](#_Toc85978745)

[設計限制 11](#_Toc85978746)

[技術限制 11](#_Toc85978747)

[介面需求 12](#_Toc85978748)

[使用者介面需求 12](#_Toc85978749)

[外部介面需求 12](#_Toc85978750)

[內部介面需求 12](#_Toc85978751)

[功能性需求 13](#_Toc85978752)

[非功能性需求 13](#_Toc85978753)

[效能需求 13](#_Toc85978754)

[測試需求 14](#_Toc85978755)

[其他需求 14](#_Toc85978756)



## 目的

藉由本學期修習的資料庫系統課程，為了更加理解資料庫的運行及流程的設計，

並且將過往學習到的網頁程式設計、網際網路技術與應用和各項程式語言的基礎學以致

用，透過分工合作完成使用資料庫的系統，因此我們決定開發一個完整的電腦配備購買平台

「霹靂卡霹靂拉拉波波莉娜貝貝魯多」，此系統能讓使用者可以在網路上進行選購電腦配備，藉由畫面精緻且簡易上手的介面，讓大家擁有賓至如歸的服務。

## System Functionality 系統功能

### [User] 系統可以認證使用者

* 系統需要擁有三種使用者，各有不同的權限
  + 顧客 (customers)：搜尋 (search)、購買 (purchase) 商品
  + 工作人員 (staff)：管理商品，像是查詢 (query) 商品、更新 (update) 商品、處理訂單資料
  + 系統管理者 (administator)：創建或管理帳戶
* 使用者可以登入、登出系統
* 系統需要以帳號、密碼驗證登入
* 系統需要可以註冊顧客帳戶

### [Customer] 顧客可以搜尋商品、購買商品、查詢歷史訂單

* 顧客可以查詢商品
* 顧客需要有購物車可以新增/移除商品
* 系統需要在顧客送出訂單時計算總金額，訂單會在顧客選擇商品並確認付款後完成
  + 系統總金額為全部的商品價格加總，包含稅金以及運費價格，若有達成條件可以使用折扣

### [Staff] 工作人員可以管理商品、處理訂單請求

* 工作人員可以管理商品，包含新增 (insert)、更新 (update)、查詢 (query)、刪除 (delete) 商品
* 工作人員可以處理訂單並更新訂單的狀態

### [Staff] 工作人員可以對訂單或商品訂定折扣方案

* + 工作人員可以管理折扣方案，包含新增 (insert)、更新 (update)、查詢 (query)、刪除 (delete) 折扣方案
  + 系統需包含三種折扣方案，包含運費折扣 (shipping)、季節折扣 (seasoning)、特殊活動折扣 (special event)，工作人員可以定義任意一種類型的折扣方案

### [Staff] 工作人員可以產生系統的報表或統計數據

* 工作人員可以生產某天 (daily)、某週 (weekly)、某月 (monthly) 的銷售統計報告

## Contained Information 系統資訊需求

### 使用者

* 有效的使用者
  + loginid
    - qunique
  + name
  + password
  + email
  + address
* 資料庫的初始狀態需包含全體工作人員 (staff) 和系統管理者 (adminstrator)
* 系統管理者也可以是工作人員

### 商品

* 商品資訊需包含：
  + productid
    - unique
* name
* description
* other attributes
  + product picture
  + price
  + quantity in stock

### 商店

* 商店資訊需包含：
  + storeid
    - unique
* name
* address
* contact email
* phone

### 訂單

* 訂單資訊需包含：
  + orderid
    - unique
  + time
  + customer
  + name
  + price of each purchased item
  + current status of the order
    - received
    - processing
    - shipping
    - closed
  + discount code
    - unique

### 銷售報告(sales report)

* 銷售報告應包含：
  + date
  + store name
  + number of orders
  + total amount of sales

## 系統名稱

本專案範圍包含建置下面主系統與各項子系統，主系統為：  
線上商場系統(Web-based Mall Business System,WMBS)  
各子系統分別為：

* 商場管理子系統(Mall Management Subsystem, MMS)
* 商品管理子系統(Commodity Management Subsystem, CMS)
* 商品呈現介面(Product Display Interfaces, PDI)
* 會員管理子系統(Account and Member Management Subsystem, A)
* 購物車子系統(Shopping Cart Management Subsystem, SCMS)
* 財務管理子系統(Financial Management Subsystem, FMS)
* 商品搜尋推薦管理子系統(Search and Recommend Management Subsystem, SRMS)  Cathy
* 資料庫子系統(Database System, DBS)

## 概觀

評估現代網頁趨勢後，選擇前後端分離，便於讓開發人員能夠專心在同一部分，不必  
因為後端的改動導致前端必須更動，耦合性降到最低，藉由這次資料庫課程，學習資料  
庫表格設計、需求書撰寫以及如何開發一個系統。  
因此，前端部分我們選擇 Vue.js 作為前端框架，相對於其他框架中，在開發上較為新手友善且便捷，也為了讓系統開發能有一套規則，維護上更為便利，故選擇 Vue.js 作為前端框架。  
後端使用 Go 語言 + net/http 框架，開發社群相當活躍。  
資料庫選擇 PostgreSQL 作為 DBMS，PostgreSQL 在高併發下能正常工作，也是相容 SQL 語言，相較於 MySQL、MariaDB，PostgreSQL 擁有優勢。

## 系統

### 系統描述

本系統主要分為七個部分，分別為商場管理子系統（Member Management Subsystem, MMS 1.1.0）、會員管理子系統（Account and Member Management Subsystem, AMMS 1.3.0）、購物車子系統（Shopping Cart Management Subsystem, SCMS 1.4.0）、商品管理子系統（Commodity Management Subsystem, CMS 1.5.0）、財務管理子系統（Financial Management Subsystem, RS 1.6.0）、商品搜尋推薦管理子系統（Search and Recommend Management Subsystem, SRMS 1.7.0)、資料庫子系統(Database System, DBS, DBS ) 。

### [系統架構圖](https://i.imgur.com/6rQcolz.png)

## 操作概念

### Scenario 1：

訪客使用者操作概念（Guest Operational Concepts）訪客使用者經由線上商場系統（WMBS）瀏覽商品，可以使用商品搜尋推薦管理系統（SRMS）查詢商品，選購商品加入購物車系統（SCMS），結帳前須申請會員（AMMS）。

### Scenario 2：

一般會員操作概念（Member Operational Concepts）會員使用者透過會員管理系統（AMMS）登入為一般會員身分，經由線上商場系統（WMBS）瀏覽商品，可以使用商品搜尋推薦管理系統（SRMS）查詢商品，選購商品加入購物車系統（SCMS）結帳。

### Scenario 3：

商場管理者操作概念（Manager Operational Concepts）使用者透過會員管理系統（AMMS）登入為商場管理者身分，除了擁有一般會員的功能，使用商品管理系統（CMS）編輯該商場所上架的任何商品。

### Scenario 4：

商場管理者操作概念 （Administrator Operational Concepts）使用者透過商場管理系統（MMS） 制定優惠券及能夠更改所有一般會員的權限，且能夠透過財務管理系統（FMS）查詢所有店家的營業額，進而分析商場的財務狀況。

## 設計限制

* 使用 PostgreSQL 作為 DBMS
* 使用 Vue.js作為前端框架
* 使用 Go net/http 作為後端框架

## 技術限制

* 會員認證(OAuth2 串接 Facebook 、 Google)
* CDN(Cloudflare)

## 介面需求

### 使用者介面需求

* 商品瀏覽介面
* 購物車介面
* 搜尋介面
* 會員登入及註冊介面
* 訂單查詢介面
* 商品管理介面
* 財務報表介面
* 商家管理介面

### 外部介面需求

* 使用者使用瀏覽器通過 HTTPS 通訊協定與 WMBS Web-server 通訊，並發送 API 請求給後端
* 後端 server 通過 Caddy 反向代理 Go 的 Web API 服務，存取本系統

### 內部介面需求

* GRPC
* SCMS 和 DBS 間必須能傳送與接收使用者購物車資訊
* AMMS 和 SCMS 間必須能傳送與接收使用者所選取之商品資訊
* SRMS 和 AMMS 間必須能傳送與接收使用者感興趣及追蹤的商品資訊
* SRMS 和 DBS 間必須能傳送與接收使用者追蹤清單及感興趣商品的相關資訊
* SRMS 和 AMMS 間必須能傳送與接收使用者感興趣及追蹤的商店資訊
* SRMS 和 DBS 間必須能傳送與接收使用者追蹤清單及感興趣商店的相關資訊
* DBS 和資料庫間必須能傳送與接收資料
* MMS 和 FMS 間必須能傳送與接收總體營業額的相關資訊
* FMS 和 DBS 間必須能傳送與接收財務報表的相關資訊
* AMMS 和 DBS 間必須能傳送與接收各種使用者登入資訊
* MMS 和 AMMS 間必須能傳送與接收商場管理員登入的相關資訊
* CMS 和 DBS 間必須能傳送與接收商品的相關資訊
* MMS 和 DBS 間必須能傳送與接收優惠券的相關資訊

## 功能性需求

* 可註冊會員
* 可搜尋與推薦喜好商品
* 可添加想要的商品進購物車
* 購物車內商品能夠結帳
* 顧客可以查看歷史訂單並且重新訂購
* 訪客依舊能將商品加入購物車
* 訪客將購物車內商品結帳前須先加入會員
* 商場管理員擁有封鎖顧客的權限
* 商場管理員能夠查看總體營業狀況
* 商場管理員能夠釋出優惠卷
* 商場管理員能夠建立與管理其他使用者權限

## 非功能性需求

### 效能需求

* 資料表的設計應正規化、節省空間。
* 使用者瀏覽時，頁面讀取應不大於 5 秒。
* 使用者搜尋時，搜尋時間應不大於 3 秒。

### 測試需求

* 每個子系統功能都應被測試過。
* 提供使用者GUI介面的測試。
* 提供各式平台使用系統的測試。
* 系統維護時，應有故障頁面通知使用者。
* 須使用 wrk 通過壓力測試。

### 其他需求

* 使用 kubernetes 將各個子系統分成獨立的 service