

壹、點餐系統前端

該專案以 [HTML + JQuery + bootstrap](#) 撰成，原始碼：<https://bit.ly/2UP0gW5>

在 <https://dinnerSystem.com> 可以完整的使用系統，測試帳號為 06610089，測試密碼為 910426。

貳、點餐系統後端

該專案以 [php](#) 撰成，原始碼：<https://github.com/dinnerSystem/Backend>

原先，這套系統只是出自好玩，並無完整的架構及系統分析，在越來越多功能的加入下，原先的架構不敷使用，而且前後端並無明顯分界，難以維護。

爾後，我決定將整個後台重構，進行系統分析，做出一個完整的 3-Tier 架構，方便系統擴充，也方便後續維護。

參、點餐系統資料庫

該專案以 [MySQL](#) 撰成，原始碼：<https://github.com/dinnerSystem/Database>

資料庫首重效能，下達高效的 SQL 指令堪稱一門藝術，為了避免資料庫吃垮伺服器資源，我以 MySQL 的 [Explain](#) 指令最佳化所有常用指令集。

[Transaction procedure](#) 以及 [deadlock](#) 也是重要的一環，不適當的使用 Transaction procedure，會造成死結頻頻發生；不適當的處理死結，會造成資料庫不一致，還可能發生「扣了錢，卻沒有顯示已繳款」的窘境。在仔細調整每個 Transaction procedure 後，[deadlock 的發生機率降低到 0.1% 以下](#)，並確保就算死結發生了，也能夠將資料庫回溯到先前狀態，避免資料庫不一致。

肆、點餐系統繳款端

該專案主要以 CSharp + dotnet Core 撰成，原始碼：<https://bit.ly/2V5nJBs>

繳款端負責處理後端發送的繳款指令，並轉送給其他服務，進行繳款。

伍、點餐系統廠商管理插件

該專案主要以 CSharp 撰成，原始碼：<https://github.com/dinnerSystem/Factory-Frontend>

管理插件提供廠商一個專門的使用者介面，方便廠商上傳菜單，也方便廠商統計今天的點單數量。

陸、點餐系統預測模型，年久失修

該專案主要以 Python 撰成，原始碼：https://github.com/lawrence910426/dinnersys_analysis

製作預測模型時，還不知道 Tensorflow 這個套件，於是自己寫了 Gradient Descent 與 Momentum method，並手動計算 back-propagation 以及梯度函數。

在撰寫程式時，意外發現特定的狀況下(訓練過程等價 Logistic Regression 時)，可以對損失函數進行「梯度下降搭配三分搜」，並大幅提高訓練效率，以下是這個發現的相關資料。

相關資料連結：https://drive.google.com/open?id=1W_3JRVgj3fX1osvHgBj0lxHPdTwoFqf5

不幸的是，這個預測模型在科展後便不再維護，年久失修，現在無法運作。

柒、點餐系統安卓前端，年久失修

該專案主要以 Java 搭配 Android Studio 撰成，原始碼：<https://bit.ly/2xUNuN2>

原先，點餐系統是個單人計畫，不打算進行團隊開發，我便自行製作一個 Android App，做為使用者終端。

不久之後，同班的白翔云同學加入團隊，在他的協助下，我們抽取出原專案的可用模組，並由他主導開發新的 Android App，再造一個更符合 UI/UX 設計的前台。

這個專案屬於點餐系統的 Prototype，在系統正式上線後，便停止維護，現在無法運作。

捌、蝸牛球分析

該專案以 Python 撰成，原始碼：https://github.com/lawrence910426/snail_analysis

將膠狀物體灌入球內，並將球體從斜坡上滾下，球體會以類阻尼運動緩緩落下，為了計算阻尼係數，我以線段樹查詢三角函數波的極值，並藉由回傳值計算阻尼係數。

蝸牛球的科展論文可以在這裡找到 <https://bit.ly/2Xc7R2O>。

玖、邦寧之塔

該專案以 Unity + CSharp + Javascript 撰成，由於年代久遠，原始碼已遺失，但是發行檔可以在 <https://drive.google.com/file/d/0B-XfHv6OeeMUVUJvbWl3dU16MTg/view?usp=sharing> 下載。

國中時期，閒來沒事，又不想讀書，便以課本為怪物，火柴人為主角，製作一款類似「洛克人」的遊戲；按下螢幕的左半邊可以跳，按下螢幕的右半邊可以攻擊，最後一關需要觸發特定 bug 才能通關。