<計畫名稱>

需求分析報告書

<單位名稱>

校訂歷史(Revision History)

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本 | 概述 | 作者 |
|  | 0.1 | 初版 (Initial version) |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. 功能性需求(functional requirement)

需求分析 - 描述性項目

* 1. …
  2. …
  3. …

由以上需求描述, 可以歸納出以下幾個事件/處理:

* 1. 客戶下訂單…
  2. 客戶取消訂單…

註:

* + 1. 描述性項目越詳細越好, 不可用情緒性的字眼。
    2. 等注意描述的邏輯性。

提示:

1. First, the whole team should conduct domain analysis(領域分析), you might want to think what the system is doing in traditional way.
2. 嘗試著鑑別有哪些事情可以由系統來自動化。
3. 嘗試著鑑別所有想要從系統取得某些東西的外部實體。(比如說:使用者到底希望系統能做什麼? 任何你想的到且希望系統能夠提供的行為, 即使它是荒謬的, 都請把它詳細地紀錄下來. 不要擔心它該如何做到。
4. 嘗試著鑑別所有想要從系統取得某些東西的內部實體。(比如說:管理者可以或是想要利用系統從事的事件)
5. 嘗試著鑑別所有需要被儲存的資料。
6. 由CRUD(Create, Retrieve, Update, Delete)四個動作來檢視系統可能發生的事件行為。
7. 想一想有沒有以資料流為導向的事件。
8. 想一想有沒有以時間為導向的事件。
9. 想一想有沒有以控制為導向的事件。
10. 利用主詞+動詞+(受詞)型式來歸納出事件.
11. 嘗試著鑑別各事件稍後會使用到的任何資訊。(利用下表來記載)

事件列表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 事件 | 觸發器 | 來源 | 活動 | 回應 | 目的地 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

提示:

* 事件:寫出造成系統去完成某事的事件
* 觸發器:引發事件的資料流
* 來源:資料來源
* 活動(使用案例):事件發生時, 系統要做什麼。活動一定要以動詞為開端. 例如:查詢 目錄(Retrieve), 更新目錄 (Update), 產生訂單(Create), 取消訂單(Delete)
* 回應:什麼樣的輸出(if any),由什麼(/哪裡)產生(系統嗎?)
* 目的地:輸出到哪裡(/給誰)

詞彙表(Glossary)

|  |  |  |
| --- | --- | --- |
| 詞彙 | 解釋 | 附註 |
| 訂單 | 訂單包含了下列資料項目 | 需要儲存 |
| 銷退單 | 銷退單包含了下列資料項目 | 需要儲存 |
|  |  |  |
|  |  |  |
|  |  |  |

提示:

* 辨別所有出現於需求描述的所有名詞.
* 辨別所有出現於事件表中的所有名詞.
* 辨別在既有的系統, 現行程序及現行報告或是表單的其他資訊.

1. 非功能性需求(non-functional requirement)
   * 1. Usability
2. 此系統對於初學者，或是對類似之系統有經驗的使用者必須容易學習及使用。
3. 對於系統的設計規格以及訓練手冊必須有詳實的文件記錄。
4. 系統的使用者必須對此系統感到滿意。
   * 1. Reliability
5. 系統當機次數必須小於3次/每年。
6. 資料庫交易失敗次數必須小於1次/每月。
   * 1. Performance
7. 系統必須能夠允許多個使用者登入。
8. 系統對於登入的過程必須在少於3秒內完成。
9. 系統對於能夠每秒同時處理500筆的交易。
10. 系統對於第3點所述不可造成系統的嚴重負荷。
11. 對於每筆交易的回應時間不可大於10秒。
12. 其他相關資料