

# 植基於雲端檔案儲存服務之檔案版本控管系統

## The File Version Control System (FVCS) based on Cloud Storage Service

李麗華  
育達商業科技大學  
資訊管理系教授  
[lhli@ydu.edu.tw](mailto:lhli@ydu.edu.tw)

李富民  
朝陽科技大學  
資訊管理系副教授  
[fmlee@cyut.edu.tw](mailto:fmlee@cyut.edu.tw)

林松科  
朝陽科技大學  
資訊管理系研究生  
[hn.s680809@msa.hinet.net](mailto:hn.s680809@msa.hinet.net)

林佳擘  
朝陽科技大學  
資訊管理系研究生  
[janny3477@gmail.com](mailto:janny3477@gmail.com)

### 摘要

本研究以雲端儲存服務為平台，提出一個檔案版本控管系統(FVCS)，讓使用者透過版本控管系統於雲端儲存服務平台下進行軟體協同開發與整合，使用者可以隨時隨地透過檔案版本控管執行程式開發工作，不需仰賴須安裝的特定主機系統。

本系統實作於台灣某食品公司資訊部門，在不擴增任何硬體設備下達到，1. 檔案空間擴增機制：利用雲端檔案儲存服務增加檔案空間，2. 檔案備援機制：利用雲端檔案儲存服務作為二次備援，3. 資料庫負載率降低：檔案與資料不儲存至資料庫，4. 跨區域檔案編輯修改：不限制於區域網路內或利用網路芳鄰連線至遠端資料夾，5. 資料查詢儲存便利性：利用雲端檔案儲存服務作為儲存媒介，隨時隨地存取作業，6. 資料安全性：利用該企業 ERP 系統登入帳號於本系統操作異動記錄，7. 降低成本。

**關鍵詞：**版本控管、雲端儲存。

### Abstract

This research is to developed a cloud-storage based File Version Control System (FVCS). FVCS allows users to co-develop and integrate programs while developing software. Participators can perform their jobs by using FVCS at anytime and anywhere.

The proposed FVCS has implemented and has applied in a Taiwan's food company. Under the premise of limited hardware, this research can achieve the following: (1) extend the requirement of file storage, (2) provide the back-up mechanism, (3) level down the database loading, (4) enable the cross-domain modifying ability, (5) facilitate the query and data storage, (6) secure the data with less redundant usage, and (7) reduce the cost of software development.

**Keywords:** Version control, Cloud storage.

## 1. 緒論

### 1.1 研究背景與動機

在軟體開發過程中，由於開發成員的流動，或是軟體規格沒有更新，導致軟體維護越來越困難。為了保護資料檔案不丟失，每位開發成員每天都在自行建立檔案備份機制，而最終儲存空間將會被大量的資料塞滿。

版本控管系統可以幫助開發專案時的多數參與者做好的記錄工作，透過保存在不同時期的檔案版本，讓工程師可以方便地縮小搜索或是比對程式異常的範圍。版本控管系統可以協助使用者管理、記錄、備份每次設計的成果，幫助加快設計流程，減少時間與人力的浪費，並可減少因為專案開發團隊人員異動而造成資料的遺失與混亂，也可提供使用者快速取得最新版本或是開發歷程中的任一個版本資料。

現行的版本控管系統大都是建立檔案庫(Repository) [14] 並採用主從式架構(Client / Server)，先將檔案集中於檔案庫，也就是某台機器上供軟體開發人員存取，該機器上必須安裝版本控管系統，才能便利的提供檔案存取的服務給使用者端。但這都僅限於在區域網路實行，且需定期備份檔案庫，以備不時之需。例如 VSS(Visual SourceSafe)或是 SVN 等版本控管軟體，必需要先安裝於區域網路內伺服器，才能透過 Web 進行版本控管功能。

雖說有了版本控管系統的支援可以提供程式開發者更為有效率的工作，但仍有遇到非上班時間使用者提出的異常狀況需馬上進行異動程式的窘境，這時除了進行遠端連線至企業電腦進行修改，別無它法。可是企業網路頻寬時時刻刻都在進行資料的傳送，遠端連線修改程式常常須花費大量時間，若是遭遇到需立即處

理的異常狀況，就會是一大難解的問題。

近年來「雲端儲存服務」已成為必備的儲存檔案方式。人們不需要帶著隨身碟，即可透過「雲端儲存服務」將資料檔案上傳至雲端，即可讓安裝有雲端儲存服務的電腦進行同步更新，不再需要帶著隨身碟走，就可享受到資料檔案隨身攜帶的便利。而「雲端儲存服務」之檔案安全性更是具備有加密壓縮，必透過帳戶管理，才能讓資料檔案能被層層保護。

若能有方法將版本控管與雲端檔案儲存服務加以整合，讓程式碼能有如隨身攜帶的便利性被儲存在雲端空間[8][10]，此空間將可被視為一個虛擬儲存空間，此虛擬儲存空間也可成為軟體開發程式碼備份空間，或是軟體規格檔案之儲存空間，且非企業儲存資源。

## 1.2 研究目的

雲端概念在網際網路上大量的被實現之時，已經有大量的雲端服務產生，其中有雲端記事、雲端音樂、雲端影像處理、雲端轉檔、雲端掃毒/軟體、雲端辦公服務、雲端行事曆和雲端儲存等等雲端服務。這些服務大多是免費的，其功能較為簡單，也有部份雲端服務需收取費用，這些服務大多以一般使用者為對象，也通常與企業核心營運系統較無關係。

雲端服務已讓使用者帶來許多的便利性，其中的雲端儲存服務更是有如隨身攜帶般，只要有電腦、有網路，檔案資料隨時可以存取。使用者無需耗費時間、人力、物力、金錢自行架構私有雲建立雲端服務，就能免費利用雲端服務提供者所提供的雲端服務。對於使用者來說，雲端就是能以小小的資訊服務成本，獲得高便利、高迅速的雲端儲存服務。

本研究之目的主要在現有的雲端儲存服務下建立檔案版本控管，利用這個系統不受限於區域網路，透過 Web Servicer 進行版本控管機制，串連雲端儲存服務將資料檔案存取於雲端，讓程式碼有如隨身攜帶之便利，使軟體開發者能不在公司的環境下，隨時隨地進行軟體開發，並將開發歷程納入版本控管系統內。

主要欲達成之任務如下：1. 檔案空間擴增機制，利用雲端檔案儲存服務增加檔案空間，2. 檔案備援機制，利用雲端檔案儲存服務作為二次備援，3. 資料庫負載率降低，檔案與資料不儲存至資料庫，4. 跨區域檔案編輯修改，不限制於區域網路內或利用網路芳鄰連線至遠端

資料夾，5. 資料查詢儲存便利性，利用雲端檔案儲存服務作為儲存媒介，隨時隨地存取作業，6. 資料安全性，利用該企業 ERP 系統登入帳號於本系統操作異動記錄，7. 低成本。

## 1.3 研究範圍

本研究因軟體工具、硬體設備上受限於經費與實作規模運作下，將利用開發工具自行架構檔案版本控管系統，並使用現有免費的雲端儲存服務 Dropbox，開發工具使用 Visual Studio.NET，檔案格式為記事本(Notepad)可開啟編輯之文字檔，檔案版本控管服務範圍為企業資訊部門人員。使用者端設定以 Windows 平台作業系統為實作環境，透過 Web Service 進行網路傳輸。本研究之成果將可提供使用者端檔案存取之服務。本系統實作所儲存之程式原始碼的類型為 Delphi 開發工具。

在雲端技術與版本控管的相關研究領域中，會有許多相關的理論技術面的涉及與其所蘊含的知識與內容，本研究僅對版本控管系統與雲端儲存結合會引用到的部份加以介紹與討論，其它相關部份知識，不在本研究探討範圍之內。

## 2. 文獻探討

### 2.1 版本控管

隨著電腦技術越發精進，很多設計過程大都是反覆、測試、溝通等流程，並且要分階段進行，因此透過記錄在設計過程中的任何異動歷程，並為每次異動編上序號，便形成一種簡單的版本控管系統。

版本控管能提供專案的軟體開發者，將程式碼還原至先前任一版本的選擇，此選擇尤其在開發過程中遇到決策異動等狀況特別重要。因有保存程式碼版本，讓軟體開發者在任何時間都可以追蹤以前的程式碼版本，並進行比較、優化，找尋最佳設計方案。

很多人開始軟體開發的時候，並沒有使用版本管控系統[16][17][18]。一但專案是多人共同開發進行時，缺少了集中的儲存地，就會使得軟體開發成員人人都有一份各自建立的程式原始碼。由於軟體開發過程中，人人都可能於時間點的不同對同一份程式原始碼進行編輯修改，如何能產生一份可以容納所有軟體開發成員編輯修改又不互相衝突的版本資訊，就是一

個困難的事情。因此需要對專案進行版本控制並加以管理，蔡煥麟[14]也提出版本控管系統可帶來以下幾點好處：

- (1)可隨時還原至歷史版本。
- (2)多人同時修改一份程式碼，不會造成相互覆蓋的狀況。
- (3)修改歷程皆會被記錄。

版本控管系統[5][6]的基本操作大致有匯入(import)、取出(check out)、上傳(check in)等動作。一般來說，在初次建立檔案庫時，可能會採取「匯入」的操作，將專案檔案直接新增至檔案庫中並納入版本控管中。而建立工作複本時，則會利用「取出」的操作，將檔案庫中的目前最新內容取出，並且放置到開發者本機中。當開發者完成了修改編輯動作之後，就必須將修改後的專案檔案「上傳」回檔案庫中。每次上傳時，上傳者必須記錄「上傳日誌(Commit Log)」來描述本次所上傳的專案檔案之原由或異動內容。

目前最常被使用且討論的版本控管系統有CVS (Concurrent Version System, 本文以 CVS 稱之) [11][15]及 VSS(Visual SourceSafe, 本文以 VSS 稱之)[12]。本研究將參用 VSS 的簡易安裝、配置、使用的優點，設計一個以 VSS 為基準的系統，開發出能進行雲端檔案儲存服務存取檔案的檔案版本控管系統。

## 2.2 雲端運算

### 2.2.1 雲端運算之定義

雲端運算技術，並不是一個創新的技術，它其實是一種虛擬的空間概念，最早提出這個概念的是美國國家標準與技術研究所(National Institute of Standards and Technology, NIST)所定義規範出來[3]。雲端運算主要運用的方式，乃是結合運用龐大電腦科技的運算技術所研發出一種網路互動模式，這種網路互動模式，也促使許多 IT(Information Technology)業者，提供多樣的雲端服務平台[3]。例如不再需要透過 Office 套裝軟體，也可在網路世界編輯文件檔案或是試算表；亦或不再需要透過美工套裝軟體，也可在雲端提供的美工服務設計圖片。這些雲端服務已大大的改善了我們在生活及工作上的便利。下列將整理學者所提出的雲端運算定義[9]，表 1 所示：

表 1 雲端運算的定義表[9]

公司名稱	定義說明
Google	雲端運算是許多應用程式與資料的整合，透過任何行動裝置（智慧型手機、平板電腦、筆記型電腦），使用 IE 瀏覽器，在雙向雲端服務（網際網路）之間，相互通連以及資料交換。
IBM	雲端運算是資料分享在網路平台上的概念，使用者只要注意雲端服務層所提供的服務即可，對於雲端基礎設備，可以忽略不去注意，因為設備會有其它雲端系統建置廠商來處理。
IDC (國際數據資訊公司)	雲端運算是一種 IT 運算整合的網路平台，這個平台可以被提出命令要求與回應相關命令，根據命令的請求訊息，系統就會提供相關雲端運算的服務。
Microsoft	雲端運算是一種資料集中供應中心，透過一套大型系統，可以供個人用戶或群體用戶使用。
NIST	雲端運算是一種模式，可以隨時隨地透過網路存取雲端平台的服務，包括：網路、伺服器、儲存裝置、應用程式、各種應用服務等。
Wikipedia	雲端運算是透過虛擬化技術，利用彈性調整虛擬化的概念，提供給不同需求的使用者。這種方式，使用者可以不用再瞭解如何建置與維護基礎設施。

### 2.2.2 雲端運算的特性

雲端運算服務，主要是運用大規模的運算概念，加上它所具有虛擬、動態、延展、管理、儲存等特性，能夠在網際網路上被運用[2]。雲端運算的運作模式並不是在單一的電腦伺服器上，而是全部存放在遠端伺服器主機端，使用者可以運用各種行動設備，結合無線網路或 3G 行動上網技術，連結至網際網路的雲端運算平台，進行各式應用服務，這個服務也就是所謂的「雲端運算服務」。雲端運算服務，透過虛擬的空間概念與系統的整合應用，提供各種多樣的服務內容，雲端運算的服務概念圖[1]，如圖 1 所示：

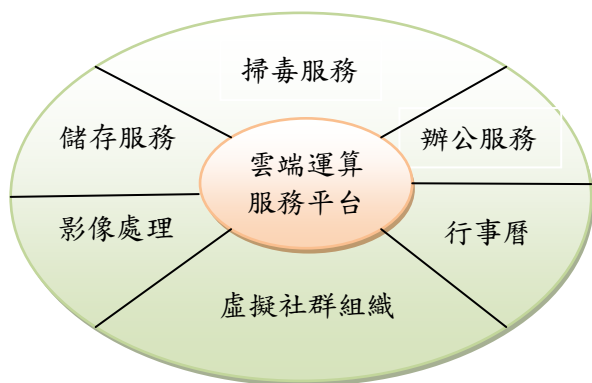


圖 1 雲端運算的服務概念圖[1]

根據雲端服務的模式，又可分成三種雲端運算的應用方式[4]，包括：私有雲端服務、公有雲端服務、混合雲端服務等，以下將詳細介紹說明。

- (1)私有雲端服務 (Private Cloud)，係指一個專屬個人或企業且獨立的雲端運算服務環境，在這個環境當中，僅存放個人或企業隱私的機密資料內容。例如：多媒體檔案（圖片、影片、音樂）、個人資料檔案、通訊錄檔案、行事曆等、企業內部資料檔案、程式原始碼等，也就是唯有私有雲端運算服務的用戶，才能管理與使用相關的服務。
- (2)公有雲端服務 (Public Cloud)，係結合所有雲端的服務機制，以提供免費的有限功能或是付費的專屬功能服務一般社會大眾或大型企業集團來使用。公有雲端服務，可以有效降低管理與建置成本，並有效提升企業效率。但是，對於網路資訊安全的責任，也就更必須要加強提升，這樣才能夠讓使用雲端服務的使用者，可以更安全的使用雲端運算的服務。
- (3)混合雲端服務 (Hybrid Cloud)，係綜合上述兩種雲端服務（私有雲端服務、公有雲端服務）的架構，一併結合成為獨立的雲端運算平台。大部份企業因為網路資訊安全的關係，傾向將資料庫或檔案存放在自己建置的「私有雲端服務」的平台上面，然後再依據雲端服務性質的不同，各別應用到「公有雲端服務」的平台上。此種透過雲端應用的方式，可以更彈性使用公有雲端的服務，並且彈性應用雲端的資源，與即時掌握資料的維護與管理；對企業經營來說，統籌並用「私有雲端服務」與「公有雲端服務」的功能，所衍生出的「混合雲端服務」，真正能夠讓雲端運算服務的基礎概念被執行與應用。

經由文獻資料，本研究統整雲端運算的優點與缺點，優點包括：1. 較低的使用成本，2. 改善效能，3. 改善不同作業系統間的相容性，4. 更易於集體合作，5. 消除特定裝置的限制。缺點包括：1. 需要不間斷、連線品質好的網路環境，2. 儲存資料之安全性未必受保障。

## 2.3 雲端儲存

雲端儲存[7][8][10]是一種網路線上儲存 (online storage) 的模式，即把資料存放在通常由雲端儲存服務廠商代管的多台虛擬伺服器，而非放在專屬的伺服器上。以往企業將儲存設備置放在公司內部的機房與企業主機相連接，因此需要建構一套完整的軟硬體，包含伺服器、儲存設備、網路設備，以及所有相關應用的軟體。此外，也需要專業人員定期的維護管理，故須耗費大量成本及人力資源。但隨著檔案 e 化的程度越來越高，企業內部的資料大幅增加，再加上政府法規要求下，客戶每一筆資料都必須留有完整的資料紀錄。各企業在人力預算有限的條件下，要如何處理龐大的資料並做好備份成了最大挑戰，於是雲端服務成了各企業資訊管理最佳的解決方案。

### 2.3.1 雲端儲存服務 Dropbox

Dropbox，一般通稱為網路硬碟、網路磁碟機、網路隨身碟等，但以上這些名稱並不能真的反應 Dropbox 的精神。其服務如下：

#### (1)檔案儲存及同步 (File Sync)

- 同步各種類型及大小的檔案，包含圖片、影音、辦公文件等。
- 同步 Windows, Mac, Linux 等 OS 電腦及 iPhone, iPad, Android 手機等透過無線寬頻網路或 3G 電信網路技術，連結至網際網路之桌上、行動裝置。
- 當偵測到有新增檔案或是檔案內容有更動時會自動同步於連結 Dropbox 之各式裝置。
- 就算是離線還是可以使用本機 Dropbox 資料夾的檔案，一旦網路恢復連線，檔案就會自動同步。
- Dropbox 檔案有續傳 (resume) 的功能，會從傳輸中斷的地方繼續而不會讓使用者重新上傳。
- 有效率的同步-只會傳送檔案變動的部份而不是整個檔案，可以節省時間及頻寬。

- 可以設定上傳/下載頻寬，不會佔滿網路頻寬。

#### (2) 檔案共享 (File Sharing)

- 共享資料夾允許多人協同使用一組的檔案。
- 其它使用者對檔案的更動可即時看到。
- 提供「Public」資料夾可以公開給其他人連結檔案。
- 控制權限存取共享資料夾（包括可以踢除某人並刪除在其電腦中的共享檔案）。
- 由相簿資料夾自動產生共享的相簿。

#### (3) 線上備份 (Online Backup)

- 自動備份檔案。
- 反刪除 (Undelete) 檔案及資料夾。
- 取回舊版本的檔案。

#### (4) 網頁存取 (Web Access)

- 新增，編輯，刪除，重新命名等等，與在電腦桌面操作一樣。
- 搜尋整個 Dropbox 中的檔案。
- 「Recent Events」彙整在 Dropbox 中的所有活動。
- 可設定共享資料夾並邀請朋友參與。
- 回復檔案先前的版本或是救回刪除的檔案。
- 觀看由 Dropbox 自動幫相片產生的相簿。

#### (5) 安全及隱私 (Security & Privacy)

- 只有邀請的人才能看到共享資料夾。回復檔案先前的版本或是救回刪除的檔案。
- 所有的資料傳輸採用加密(SSL)的通訊協定。
- 所有儲存在 Dropbox 伺服器的檔案都用 (AES-256) 256 位元加密過，只有正確的帳號密碼才能存取。
- Dropbox 網站及用戶端程式都已加強防駭功能。
- Dropbox 員工不能看任何使用者的檔案。
- 線上存取檔案需要帳號及密碼。
- 公開的檔案只能由使用者給的隨機連結網址取用。公開資料夾是無法瀏覽或搜尋的。

#### (6) 行動裝置存取 (Mobile Device Access)

- 隨時隨地存取。
- 在 iPhone, iPod Touch, iPad, Android 手機等可觀看檔案。
- 下載檔案以供離線觀看。

- 用手機照相及拍影片並自動上傳與同步。
- 分享檔案連結。
- 觀看互動式的相簿。
- 下載檔案保持最新狀態。

本研究欲減少企業內部備份檔案之儲存空間並節省成本，故將運用 Dropbox 來做為版本控管的儲存空間。

### 3. FVCS 檔案版本控管架構設計

本研究欲提出一個植基於雲端檔案儲存服務之檔案版本控管架構，本研究將提出此架構之說明與設計概念，並將此一系統名稱命名為 FVCS (File Version Control System)。以下說明 FVCS 之目的、架構與設計方法。

#### 3.1 FVCS 目的

FVCS 設計想法為應用現有雲端檔案儲存服務來儲存資料並同時作為備援功能，透過同步更新之服務機制的主從式架構，以設計出低建置成本、維護成本、移植成本之雲端檔案版本控管服務系統為目的。

以一般企業員工所採用之作業系統而言，大抵 Windows 平台之作業系統為主要來源，而目前運用雲端檔案儲存服務最為被廣泛討論的主要有 Dropbox。本研究期許能以建構低建置成本、維護成本、移植成本之檔案版本控管系統為基準，因此以經常被使用的 Windows 平台主機做為軟體開發工作平台，利用本研究所設計之檔案版本控管系統將檔案透過 Web service 上傳或備份至雲端檔案儲存 Dropbox 中，在 Dropbox 的雲端儲存服務下，亦可將檔案下載或分享至使用者端主機進行修改編輯。

#### 3.2 FVCS 架構

本研究所建置的 FVCS 檔案版本控管設計概念採用主從式架構，將分為三個層次，分別為應用層、控制層、儲存層，如圖 2 中所示。主要是讓使用者利用使用者端介面提出需求透過 Web service 來溝通伺服器端，再將檔案上傳下載同步至雲端檔案儲存服務。

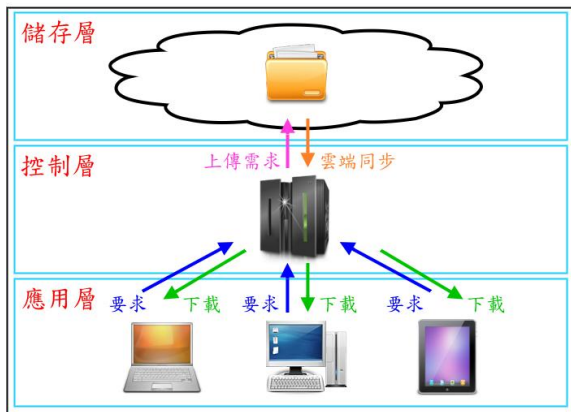


圖 2 FVCS 雲端檔案儲存服務概念圖

以下為 FVCS(如圖 2)之各層說明：

- 應用層：此階層主要為使用者透過前端互動介面，設計互動介面以應用程式類型提供使用者端對檔案版本控管系統申請儲存、查詢、讀取、比對功能。應用程式介面通過網路連線將使用者端需求透過 Web service 傳送到控制層，由控制層執行或處理從儲存層存取所需之資訊或檔案。
- 控制層：透過 Web service 接受應用層之使用者前端的申請需求，如查詢、上傳、下載、比對、版本控管等對儲存層做檔案的存取動作，使應用層之使用者執行或處理資料與檔案的修改編輯查詢比對等應用，並記錄其各種異動行為。為使達成控制層主要功能，必須事先建立版本資料初始資訊化，透過版本資料初始資訊化的動作，產生一個檔案版本資料庫，此為版本控制系統的核心，利用此檔案版本資料庫來新增版本與使用者異動行為的資料流向。
- 儲存層：擔負檔案儲存與備援之任務，由控制層接受應用層之使用者需求，經由控制層存取回饋相關資訊於應用層之使用者前端。接收自應用層需求經由控制層上傳檔案後，依照檔案版本控管系統之規則，將檔案儲存至雲端檔案儲存服務；當接收到應用層需求透過控制層取回檔案後，將雲端檔案儲存服務內之相關檔案傳遞給應用層之使用者。因本階層採用雲端檔案儲存服務，當透過 Web service 存取檔案時，不用透過層層關卡如防火牆機制，造成無法存取檔案。

### 3.3 雲端檔案儲存服務 FVCS 設計

雲端檔案儲存服務乃係透過雲端服務業者

提供之軟體進行架構與伺服器溝通，而其檔案版本控管系統軟體開工具採用 Visual Studio 2008，資料庫採用 Microsoft SQL Server 2008 Express[13]作為資料索引中心。

本研究以實作雲端檔案儲存服務之檔案版本控管系統架構，如圖 3 為設計內容，控制層由控制主機與索引(資料庫)主機構成，此為本系統之執行處理核心，控制主機負責接收與提供使用者端服務，並將檔案索引資料保存於索引主機作為版本控管與使用者異動行為之資料流，應用層由管理者與使用者透過資料流，分析該檔案的歷史與異動狀況。儲存層負責儲存實體檔案，由雲端服務業者提供之軟體與伺服器進行溝通與同步。

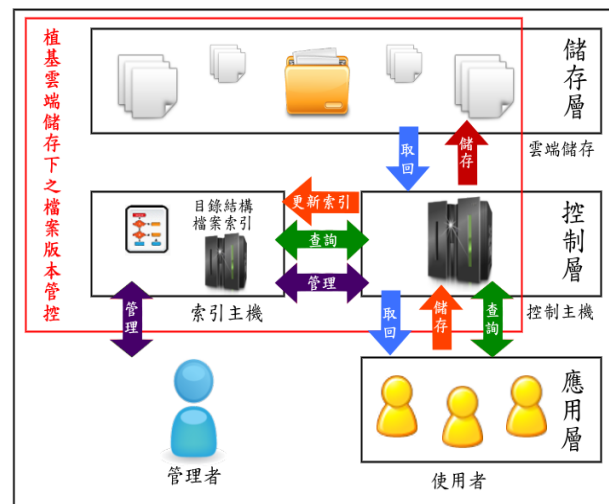


圖 3 雲端檔案儲存服務 FVCS 架構圖

### 3.4 雲端檔案儲存服務 FVCS 流程

在圖 2 和圖 3 當中，已經介紹有關於 FVCS 概念與架構圖。圖 4 FVCS 流程中，將介紹本系統應用層與控制層、儲存層之細部流程圖。

- (1)使用者連線：當使用者啟動使用者端應用軟體時，應用軟體會先與控制主機進行比對使用者資訊是否正確，成功登入系統後，方可使用系統提供如查詢、新增、編輯與比對等功能。
- (2)使用者檔案新增：使用者透過軟體新增專案檔案功能，經由控制主機於雲端資料夾新增專案資料夾及其檔案，並將其相關資訊記錄於資料庫後，再透過控制主機上雲端檔案儲存服務提供之軟體同步於雲端。
- (3)使用者檔案更新：使用者透過軟體 Check in 功能，經由控制主機對異動資料表新增異動

記錄，並將控制主機雲端資料夾內舊版本專案資料夾以專案名稱加入年月日時分秒規則更改資料夾名稱，與使用者端新版本專案資料夾及其檔案新增至控制主機雲端資料夾，後再透過控制主機上雲端檔案儲存服務提供之軟體同步於雲端。

- (4)使用者查詢：使用者透過軟體提供之各項查詢功能，依使用者需求向控制主機發出申請，控制主機會向資料庫查詢目錄與檔案索引相關資訊，並回饋至使用者端訊息。
- (5)比對：使用者透過軟體經由控制主機上雲端檔案儲存服務提供之軟體將檔案下載回使用者端。進行歷史版本比對，可了解歷史版本修改原由與異動過程、異動者等資訊。

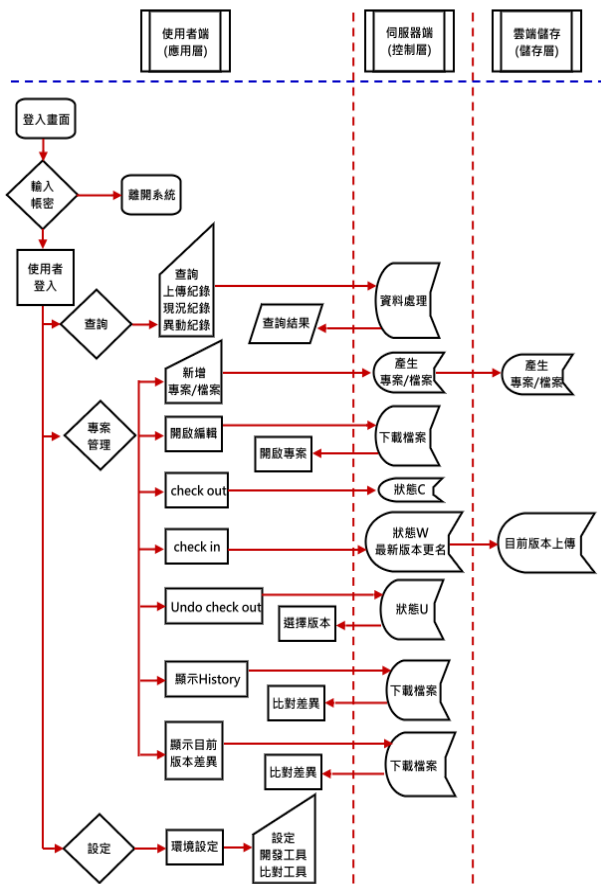


圖 4 FVCS 流程

### 3.5 索引資料庫設計

索引主機資料庫主要目的在存放雲端檔案儲存服務之檔案資訊、目錄結構、主機位址與使用者權限、帳號等等。使這些檔案資料，讓使用者能有效查詢到檔案之修改歷程，了解不同版本的進行。

- (1)資料庫系統：本研究所採用之資料庫系統為

Microsoft SQL Server 2008 Express[13]，屬小型關聯式資料庫系統，免費提供小型軟體專案需求，唯限制單一資料庫檔案大小為 10GB，只支援單一 CPU 與 1GB RAM。使用資料庫系統必須有硬體機器，才能安裝資料庫系統，其資料庫特性如下：

- 由 Microsoft 所開發的 SQL Server 的其中一個版本，此版本是免費且可自由轉散步(需經註冊)，並且可和商用程式一起使用的小型資料庫管理系統，它繼承了多數的 SQL Server 功能與特性，像是 Transact-SQL、SQL CLR 等，相當適合使用在小型的網站，或者是小型的桌上型應用程式，它也可以和 SQL Server 整合，作為資料庫複製(Replication)的訂閱端。
  - 資料庫的大小最高只有 10GB，這個大小的限制只有在資料檔案上，交易記錄檔則不受此限。
  - 只能使用一個 CPU 運算，這在多個 CPU 的電腦上會造成浪費。
  - 可使用的記憶體量最高只有 1GB。
  - 沒有 SQL Agent，若要做排程必須自己寫程式。
- (2)資料表設計：FVCS 索引資料庫共有 6 個資料表，如表 2~表 7，資料表目的以專案與檔案明細表為核心，提供管理者作專案的權限與編碼設定與記錄使用者操作行為及其上傳專案版本。

資料表欄位說明：

- USERAUTH(權限設定資料表)

表 2 內存放權限設定基本資料，包含專案代碼、使用者、設定顯示與 Checkout 權限。系統預設在新增專案時會產生一筆權限設定資料，預設該專案可對所有使用者顯示並執行 Check-out 動作。本資料表主要為設定為管理者操作。

表 2 權限設定資料表(USERAUTH)

資料表		vss_USERAUTH		權限設定		
項次	欄位代碼	欄位名稱	型態	長度	備註	
01	DB_NAME	系統名稱	C	32	EFP	
02	PROJ_ID	專案代碼	C	8		
03	USERID	使用者	C	20		
04	ENABLE	顯示	C	1		
05	CTRL	Checkout	C	8	管控是否可Checkout	

● Proj\_Head(編碼方式資料表)

表 3 內代表本資料表主要設定新增專案時，專案代碼開頭的命名規則，用以代表不同的專案類別，如 ECLI 類別與 ESRV 類別，分別代表 Clinet 端與 Server 端專案。

表 3 編碼方式資料表(Proj\_Head)

資料表		vss_Proj_Head	編碼方式		
項次	欄位代碼	欄位名稱	型態	長度	備註
01	DB_NAME	系統名稱	C	32	EPP
02	PROJ_GROUP	專案類別	C	8	ECLI / ESRV
03	HEAD_STR	標題字串	C	8	EC / ES

● Proj(專案明細資料表)

表 4 內代表此資料表存放專案資訊，如專案類別、專案名稱、專案負責人(建立此專案之使用者)、專案最新版本、專案最新版本上傳者、專案狀態是否是被 Check-out、專案 Check-out 之使用者等資料。在經過各種異動行為，系統會更新本資料表該專案之資訊，以提供使用者在操作檔案版本控管系統時，能即時發現專案的異動狀況。

表 4 專案明細資料表(Proj)

資料表		vss_Proj	專案明細		
項次	欄位代碼	欄位名稱	型態	長度	備註
01	DB_NAME	系統名稱	C	32	EPP
02	PROJ_ID	專案代碼	C	8	☺
03	PARENT_PROJ	母專案代碼	C	8	
04	PROJ_TYPE	專案類別	C	4	
05	PROJ_NAME	專案名稱	C	32	
06	CUR_OWNER	專案負責人	C	20	通常為專案建立者
07	CUR_BPLVER	最新版本	C	12	
08	CUR_UPBY	最新版本上傳者	C	20	
09	RMK	備註	C	255	
10	STATE	狀態	C	4	L:Checkout / N:Normal
11	LOCK_BY	Checkout之使用者	C	20	

● File(檔案明細資料表)

表 5 內代表檔案明細資料表存放專案下之檔案明細，用以提供使用者能對檔案使用滑鼠右鍵功能進行檔案歷史版本比對。

表 5 檔案明細資料表(File)

資料表		vss_File	檔案明細		
項次	欄位代碼	欄位名稱	型態	長度	備註
01	DB_NAME	系統名稱	C	32	EPP
02	PROJ_ID	專案代碼	C	8	☺
03	FILENAME	檔案名稱	C	32	

● FILE\_UP\_LOG(版本明細資料表)

表 6 內代表版本明細資料表用於使用者點擊 Check-in 功能時所產生的資訊存放之資料表。當 Check-in 時系統會顯示對話視窗提供使用者輸入本次異動之目的內容，當使用者確認異動目的內容後，即將最新版本專案資料夾以日期時間更名，並將日期時間更新於版本欄位，同時更新專案明細資料表相關資

訊。後將使用者所 Check-in 之專案資料夾透過 Web service 上傳至雲端檔案儲存服務，同時新增該專案相關資訊。另為避免無法查詢在編譯專案時產生的執行檔是由哪些專案資料夾下的檔案版本組成，本研究設計成以專案資料夾為單位做版本的控管，而不採用以檔案為單位做版本的控管。因軟體工具的不同，有些專案必須透過編譯不同的檔案來產生執行檔，這些執行檔並無相關檔案組成資訊，故無法查詢所需的資料，這在版本控管中是一大缺點，現以專案資料夾為單位做版本控管，可避免此缺點的發生。

表 6 版本明細資料表(FILE\_UP\_LOG)

資料表		FILE_UP_LOG	版本明細		
項次	欄位代碼	欄位名稱	型態	長度	備註
01	DB_NAME	系統名稱	C	32	EPP
02	PROJ_ID	專案代碼	C	8	
03	FILENAME	檔案名稱	C	32	
04	CUS_VER	版本	C	12	
05	VER_BY	上傳者	C	20	
06	RMK	異動內容	C	255	

● FileOpLog(異動明細資料表)

表 7 內異動明細資料表主要存放檔案版本控管系統之日常工作記錄。其中包含使用者新增專案、Check-out 專案、Check-in 專案、UndoCheckout 專案等異動行為，提供使用者查詢專案相關資訊。

表 7 異動明細資料表(FileOpLog)

資料表		vss_FileOpLog	異動明細		
項次	欄位代碼	欄位名稱	型態	長度	備註
01	DB_NAME	系統名稱	C	32	EPP
02	PROJ_ID	專案代碼	C	8	
03	FILENAME	檔案名稱	C	32	
04	OP_USER	操作者	C	20	
05	OP_TIME	操作時間	C	14	
06	OP_TYPE	操作類型	C	4	C: Check out / W: Checkin U: UndoCheckOut A: Create / D:Delete
07	LOCK_MACH	Checkout主機名稱	C	32	
08	LOCK_PATH	Checkout主機檔案路徑	C	255	
09	RMK	異動內容	C	255	

## 4. FVCS 實作

本研究所提之 FVCS 系統實作於台灣某食品股份有限公司之資訊部門內，因該部門本身於軟體開發時並未有檔案版本控管之機制，所有程式原始碼檔案僅儲存在伺服器空間，此空間已有大量軟體程式原始碼檔案，為使檔案版本控管機制能快速建立，故必須先進行資料初始資訊化的任務。藉由資料初始資訊化，使其對相關專案名稱、專案路徑、檔案名稱、檔案大小及版本資料等資訊產生資料庫並成為檔案版本控管機制之來源。



資料初始資訊化後，於伺服器端主機啟動檔案版本控管服務，使其在使用者端可以透過該服務進行檔案版本控管與檔案儲存於雲端檔案儲存服務。

FVCS 實作於台灣某食品股份有限公司現階段之實作流程圖如圖 5，以下就階段流程圖分別以小節作說明。

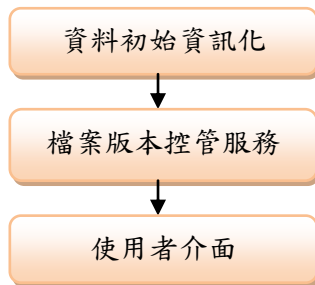


圖 5 實作階段流程圖

## 4.1 FVCS 主機端應用程式設計

### 4.1.1 資料初始資訊化

在實際使用檔案版本控管系統前，必須要將相關資料與檔案分別輸入資料庫與雲端儲存服務內，以供未來透過檔案版本控管系統存取檔案之用。本階段主要任務為初始化資訊欲作為版本控管之專案資料檔案，透過初始資訊化軟體，將專案資料一一寫入資料庫並複製檔案至雲端檔案儲存服務，等於是將專案內資料檔案資訊化。這對未來查詢檔案異動狀況或是比對版本內容不同時，因有了這個初始資訊化的動作才能有後續功能的產生。

FVCS 資料初始化可由圖 6 得出，即是設定相關資料，如來源目錄(原 Source 目錄)與目的目錄(工作 Source 目錄)、專案名稱、專案命名原則、資料庫連線字串等。經由以上相關設定後，開始執行並一一讀取原 Source 目錄下專案檔案，首先將專案檔案複製至工作 Source 目錄下，並於複製時將該專案檔案資料寫入資料庫系統，如專案名稱、檔案名稱、檔案目前版本等資訊，其中檔案目前版本設定以日期時間為版本代號。

當完成初始化資訊設定完成後，其內有之檔案版本則為初版，未來則經由檔案版本控管系統接手版本管理，無論是 Check-in、Check-out 等等異動行為，皆由檔案版本控管系統管理記錄與上傳下載檔案。本初始資訊化軟體會因初始資訊化檔案過多，也為避免耗時，初始資訊化僅執行一次建立初版內容資訊，只在搜集資

料，旨在搜集資料，而本項任務之操作人員主要為管理者。

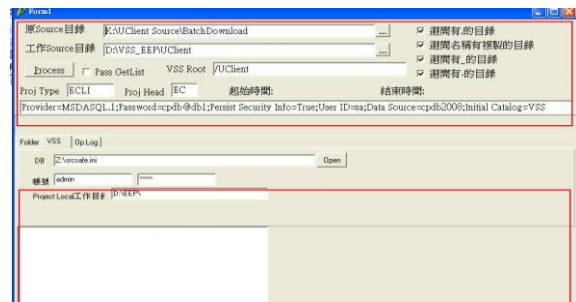


圖 6 初始資訊化軟體設定畫面

圖 6 初始資訊化軟體設定說明：

- 原 Source 目錄：原始 Source 工作目錄，初始資訊化前需整理欲將檔案版本控管之專案資料檔案，以利初始資訊化執行。若未能初始資訊化之專案資料檔案，也就是未來所建立的專案檔案資料，屆時可透過檔案版本控管系統新增專案功能，達到檔案資訊化之目的。
- 工作 Source 目錄：雲端檔案儲存服務空間，備份專案資料檔案之所在，亦即是使用者上傳下載存取檔案的檔案庫。
- VSS Root：設定匯入的 Source 在檔案版本控管系統類別下。
- Proj Type：匯入的 Projects 類別→Client 是 ECLI，Server 是 ESRV。
- Proj Head：Project 的 ID 編碼前兩碼，作為資料表之索引主鍵。
- VSS DB SQL：資料庫連接資訊，作為寫入初始資訊資料之資料庫連接資訊。

資料初始化設定完成，則接著進行初始資訊化工作，圖 7，說明初始資訊化進行時的相關畫面。左下方為專案目錄，右下方為每一專案目錄下檔案，當經由相關設定後，開始進行初始資訊化，這時初始資訊化軟體讀取到該專案目錄時，會進行專案及專案目錄下檔案複製，並將其相關資料寫入資料庫系統，如專案名稱、檔案名稱、檔案目前版本等資訊，其中檔案目前版本設定以日期時間為版本代號，以此作為檔案版本控管系統來源依據。本系統規劃兩大類別，分屬 ECLI、ESRV，各代表 Client 與 Server 端之程式碼，此兩大類別下又有不同的專案及其專案下之檔案，主要目的是區分 Client 與 Server 端之程式碼。

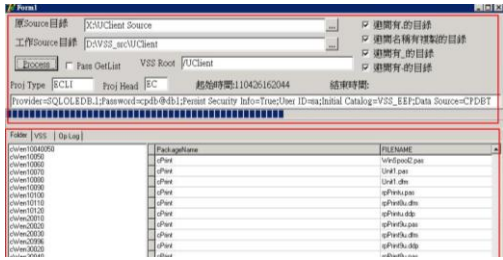


圖 7 初始資訊化軟體運行畫面

### 4.1.2 FVCS 服務

因本研究採用主從式架構，此伺服器作為資料庫與應用程序主機，檔案版本控管服務主要利用此伺服器為 Web 服務(Web service)，Web 服務是一種軟體服務，它透過標準的 Web 通訊協定及資料格式的開放式標準(例 HTTP、XML 及 SPAP 等)來為其它應用程式或異質系統提供服務或進行資料交換。其目的主要是根據接收來自使用者端需求申請，經由檔案版本控管服務確認資訊後向雲端檔案儲存服務空間做讀取或儲存檔案之異動動作，並記錄使用者端對於此次異動之目的內容與存取時間，作為下一版本之依據。本程式主要運作於伺服器主機，啟動後將無使用者操作。如圖 8，操作人員主要為管理者。

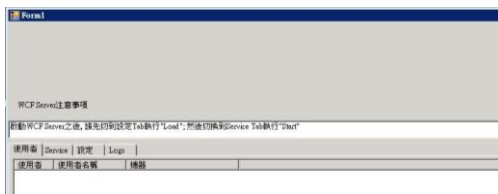


圖 8 檔案版本控管服務啟用畫面

圖 9 可看到每次開啟檔案版本控管服務，並不是直接啟用服務，而需讀取設定資料後，再啟動服務。此為避免軟體一啟動時讀取錯誤的設定資料或是設定資料已被變更(例變更為新資料庫來存取資料)導致連接失敗。故需先讀取設定資料，再啟動服務。

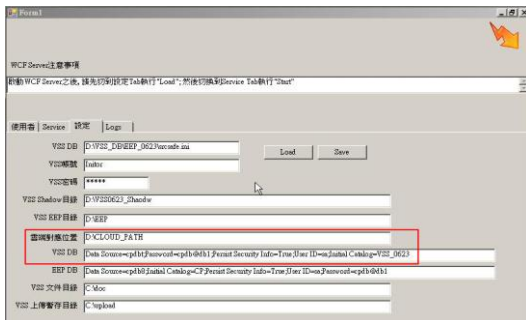


圖 9 檔案版本控管服務設定畫面

圖 9 檔案版本控管服務設定畫面說明：

- 雲端對應位置：雲端檔案儲存服務之本機指定資料夾；雲端檔案儲存服務若以 Dropbox 為例，可於本機指定 Dropbox location，此 Dropbox location 則是與 Dropbox 同步的資料夾，當上傳下載檔案時，存取於此雲端對應位置，也是將檔案同步於雲端檔案儲存服務。
- VSS DB：連線到檔案版本控管系統資料庫的連線字串。對於使用檔案版本控管系統的使用者做異動記錄，無論是新增專案、修改編輯檔案、上傳檔案等異動，都會一一記錄於資料庫。
- Save：將檔案版本控管服務設定資料儲存成副檔名 ini 的設定檔，方便檔案版本控管服務執行時，將設定檔讀入，作為啟用檔案版本控管服務的依據。
- Load：讀取檔案版本控管服務設定資料，讀取副檔名 ini 的設定檔後，待開啟檔案版本控管服務。

檔案版本控管服務設定資料完成後，即可讀取檔案版本控管服務設定資料，確認無誤即可切換頁面至 Service 頁面，啟用檔案版本控管服務，如圖 10。

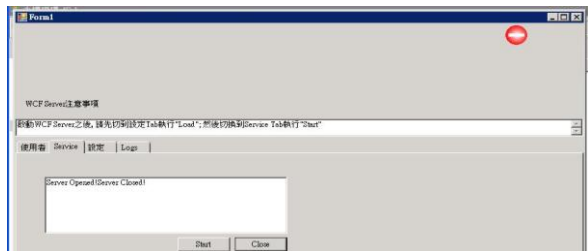


圖 10 檔案版本控管服務啟動與停止畫面

檔案版本控管服務啟用後，使用者即可登入服務並使用檔案版本控管系統，如圖 11，檔案版本控管服務啟用後登入使用者之資訊。若有不正常登出檔案版本控管服務，將造成無法登入檔案版本控管服務使用功能，管理者可透過此頁面將使用者登出服務。



圖 11 檔案版本控管服務使用者登入資訊畫面

圖 12，檔案版本控管服務記錄使用者異動之資訊，左方框顯示時間、操作者、操作訊息，右方框顯示操作訊息之 SQL 語法指令碼，藉以了解使用者實際操作行為，如對資料表進行撈取資料或是新增上傳專案等等動作。

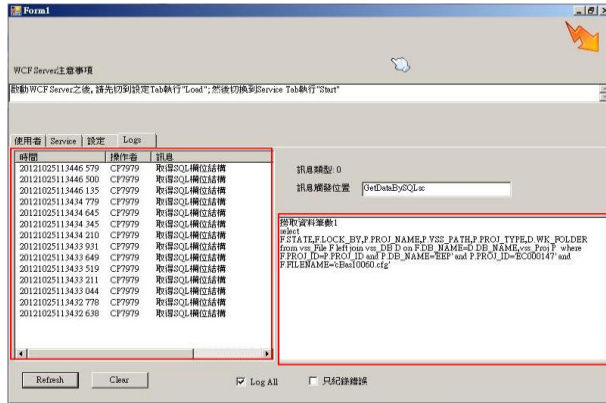


圖 12 檔案版本控管服務異動記錄畫面

## 4.2 FVCS 使用者端應用程式設計

本研究透過軟體方式將雲端服務與版本控管結合，建構一套企業內部以雲端檔案儲存服務之架構檔案版本控管系統，此系統需搭配使用者端應用介面方能發揮效益。本章節將以使用者資料交換需求開發一軟體介面，透過檔案版本控管服務溝通雲端檔案儲存服務，建置檔案版本控管，並以安全性、即時性、便利性為設計目標。因此透過本系統提供企業資訊部門檔案版本控管，預期達到以下目標：

- (1) 此軟體能提供使用者自行建立專案與檔案，上傳下載程式原始碼檔案功能。因為軟體開發會不斷的有新專案檔案，當然，新的專案檔案也必需被納入檔案版本控管系統管控，如此才不至於檔案版本控管只能控管已備份初始資訊化的專案檔案。
- (2) 當檔案已被 Check-out 或是 Lock 狀態，其它使用者只能讀取檔案，並不能修改檔案，唯有當檔案已被 Check-in 或是處於 Unlock 狀態才可以對其 Check-out 進行編輯修改，此為檔案版本控管功能。
- (3) 可比對最新檔案版本與歷史版本間之差異，此功能為方便使用者能在查找時，透過最新版本與歷史版本間的比對差異，使軟體開發使用者能了解該檔案歷史資訊。

### 4.2.1 FVCS 軟體介面

#### (1) 檔案版本管控系統使用者設定介面

本系統如同一般使用者介面，皆需輸入帳號密碼來登入系統，此登入動作也是為了本系統為了在各種異動操作時的記錄，以利在查詢版本或是其它訊息時之用。

圖 13，為增加使用者便利性與降低管理者複雜度與避免使用者記憶太多有關企業相關系統登入資訊，本系統統一以公司 ERP 系統之使用者登入帳號密碼資訊作為控管，任何使用者皆能以此檔案版本控管系統連接雲端儲存服務，進行檔案版本控管。而目前本軟體僅供企業資訊部門人員使用。



圖 13 檔案版本控管服務登入畫面

圖 14，藉由登入帳號密碼資訊後，經由 Server 端檢核登入資訊後無誤即可登入主畫面，登入主畫面後，主畫面上方分別為設定選單、查詢選單、專案選單等多項選單功能，主要是用以設定 Client 端使用者各項資訊功能，如變更帳號密碼、設定開發用工具程式開啟路徑、設定比對程式開啟路徑等等。

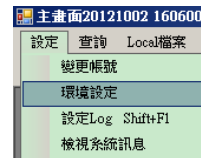


圖 14 檔案版本控管服務功能設定選單畫面

圖 15 環境設定畫面，設定各種軟體之路徑，以利開啟開發用工具程式編輯修改或開啟比對程式比對不同版本間之異動等功能。

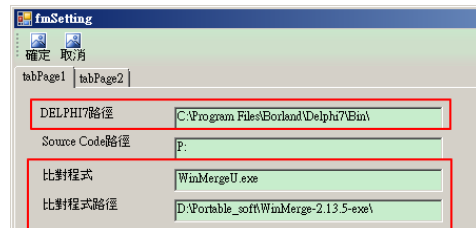


圖 15 檔案版本控管服務環境設定畫面

圖 15 檔案版本控管服務環境設定畫面說明：

- Delphi 路徑：Delphi 程式路徑，可透過本系統於主畫面上專案窗格利用滑鼠右鍵開啟功能選單點選功能使用。本功能主要是方便軟體開發者不必再花時間開啟軟體開發工具，經由本系統設定後，利用本系統開啟軟體開發工具並載入欲修改之專案檔案。
- 比對程式、比對程式路徑：本系統因需使用比對功能，現採用 Free 軟體來作為比對使用，在使用者安裝比對軟體後，需在本系統之環境設定下設定有關比對軟體與程式編輯軟體之路徑，以程式編輯或是比對原始碼時開啟軟體之用。當設定完成後，可透過本系統於主畫面上專案窗格利用滑鼠右鍵開啟功能選單點選功能使用。本功能主要是方便軟體開發者不必再花時間開啟比對工具，經由本系統設定後，利用本系統開啟比對工具並載入欲比對之專案檔案。而因 WinMergeU 這款比對軟體對於檔案格式有限制，如副檔名.txt、.asp、.aspx、.cs 等相關可使用 Notepad 軟體開啟之檔案，而無法比對 Word、Excel 等特定格式作比對功能。

因軟體開發系統大都採用主從式架構，本系統設定架構為兩大類別，分別為 UClient、UApSrv 類別，各類別下會有各別的專案目錄及其檔案。於是使用者端必需設定 UClient、UApSrv 類別的使用者端工作目錄，經由設定，系統會將取得之專案檔案資料下載至使用者端工作目錄，或是將使用者端工作目錄內專案檔案資料上傳至 Server 端與雲端檔案儲存服務。此工作目錄為使用者本機內之資料夾，上傳下載時皆對此工作目錄異動，圖 16~圖 19。為了不造成使用者困擾與麻煩，有關設定資訊只需要於首次使用本系統時設定即可，不需於每次登入本系統再次設定相關資訊。而設定資訊是為了每位使用者的工作目錄或是軟體開發工具、比對軟體的不同而設立的。至於有關連接 Web service 之功能已載入至使用者端，使用者不需設定。



圖 16 檔案版本控管服務工作目錄 UClient 設定選單

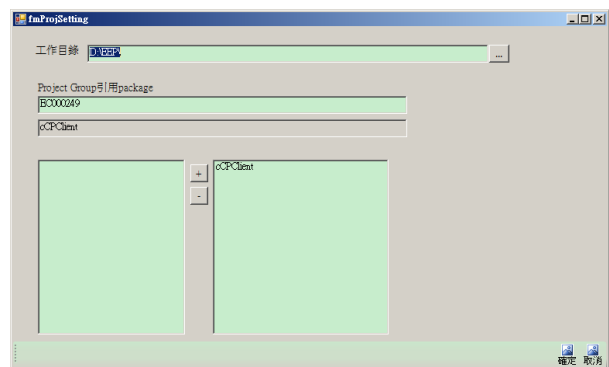


圖 17 檔案版本控管服務工作目錄 UClient 設定畫面



圖 18 檔案版本控管服務工作目錄 UApSrv 設定選單

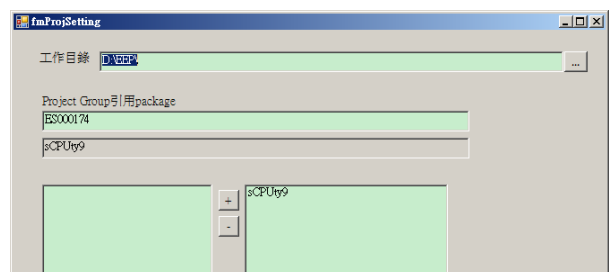


圖 19 檔案版本控管服務工作目錄 UApSrv 設定畫面

(2) 檔案版本控管系統使用者端主畫面

經由建立工作目錄與軟體開發工具、比對軟體等相關設定後，進入本系統使用者端介面的核心，圖 20 專案選單畫面，開啟檔案版本控管系統主要操作畫面。



圖 20 檔案版本控管服務專案選單畫面

檔案版本控管系統使用者端主畫面最上方為選單，下方為登入資料，左方為 UClient、UApSrv 類別下各項專案之目錄與篩選過濾條件，右方為各項專案之目錄下檔案及其版本等相關資訊。左方專案目錄右鍵功能主要作為新增專案或是 Check-in、Check-out、開啟 Dephli 編輯檔案等等功能之用，圖 21。

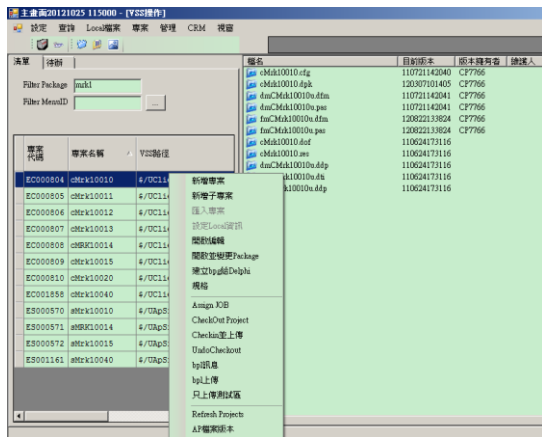


圖 21 檔案版本控管服務使用者端主畫面專案右鍵功能選單

圖 21 檔案版本控管服務使用者端主畫面專案右鍵功能選單說明：

- 新增專案：在初始資訊化時，未加入之專案檔案資料，或是當初始資訊化之後所新增的專案檔案資料，必須透過新增專案功能對其作資訊化動作並納入檔案版本控管。若有新專案需加入檔案版本控管，點選「新增專案」，即可新增新專案目錄。
- CheckOut Project：Check-out 專案，當使用者需要修改編輯該專案，系統會顯示對話視窗，要求使用者輸入本次異動修改內容，此舉是因為專案的進行是長久性的，為了避免使用者於專案上傳時遺忘當初專案新增修改的目的，於使用者 Check-out 專案時，非強制性要求使用者輸入異動內容或目的來達到日後 Check-in 時的提醒。輸入異動內容後系統必須將該專案作 CheckOut Project 並對資

料庫作記錄，代表該專案已被某位程式開發者取出用以修改編輯，而系統也會將該專案與其專案目錄下檔案資料下載至使用者端工作目錄。使用者 Check-out 之專案，僅限專案之最新版本。Check-out 後，使用者端主畫面右方檔案名稱前圖片則會變為紅色，代表檔案已被 Check-out。因為每個專案都是唯一存在性，當專案已被某位使用者 Check-out 後，其它使用者僅能取出該專案檔案資料，無法對該專案檔案做 Checkin 並上傳之動作，僅有 Check-out 專案檔案之使用者才有 Checkin 並上傳的權限。Check-out 專案功能，圖 22~圖 24。

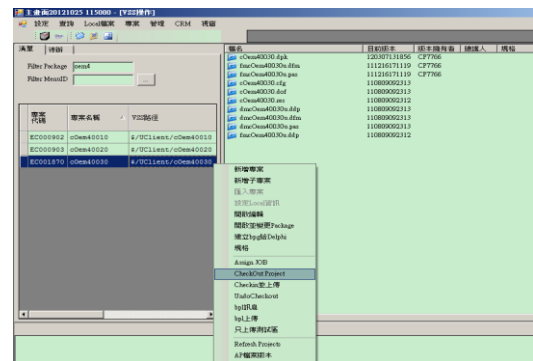


圖 22 檔案版本控管服務 CheckOutProject(1)

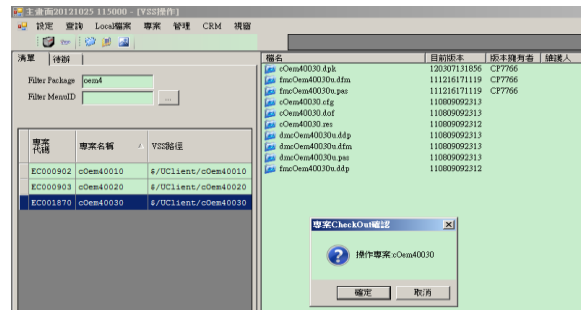


圖 23 檔案版本控管服務 CheckOutProject(2)

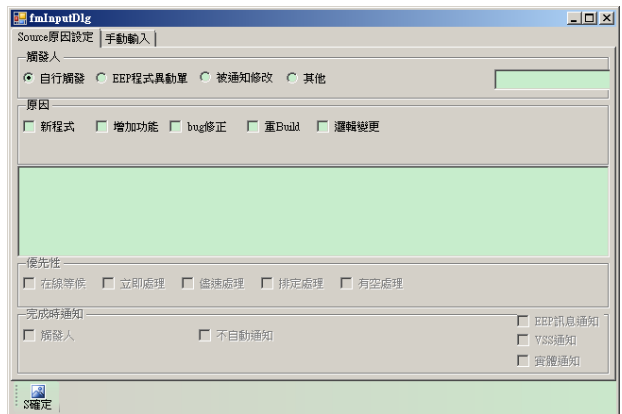


圖 24 檔案版本控管服務 CheckOutProject(3)

- 開啟編輯：透過先前使用者端之設定資訊，當使用此功能時，則會開啟軟體開發工具。本系統因係使用 Delphi 軟體開發工具，使用此功能，則會開啟 Delphi 程式用以編輯修改程式碼，圖 25~圖 26。
- Checkin 並上傳：Check-in 已被 Check-out 之專案檔案，本功能僅限已被該使用者 Check-out 之專案檔案做 Check-in 動作，並將原專案實體目錄做複製更名動作，再對已編輯修改後之專案檔案上傳至 Server 端與雲端檔案儲存服務，用以作版本區別。Check-in 後，使用者端主畫面右方檔案名稱前圖片變為藍色，代表檔案已被使用者 Check-in，並對資料庫做記錄，其它使用者才可利用 Check-out 功能對該專案內檔案進行編輯修改。
- UndoCheckout：是為避免使用者可能對專案 CheckOut 錯誤，本功能主要用以提供使用者對該專案進行 Undo 動作，讓其它使用者能對該專案 CheckOut 編輯修改。

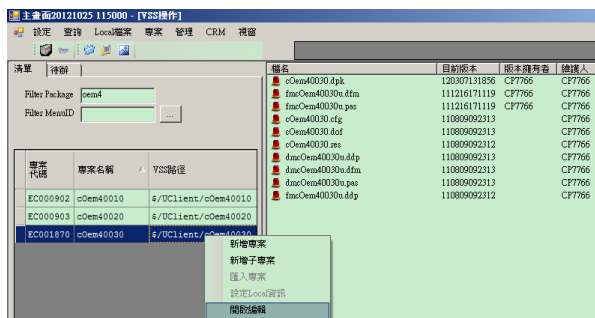


圖 25 檔案版本控管服務開啟編輯(1)

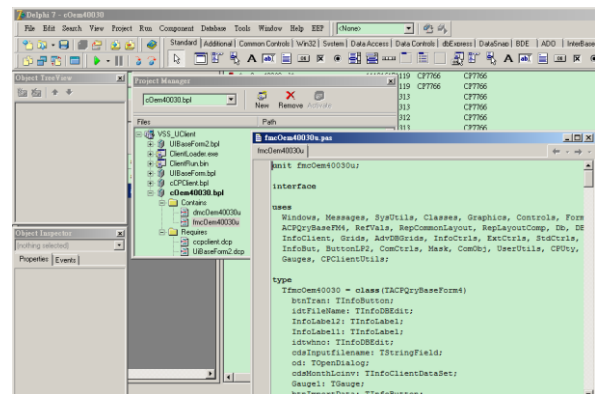


圖 26 檔案版本控管服務開啟編輯(2)

(3) 查詢

為讓使用者可快速查詢到目前專案檔案之上傳的狀況、異動內容與尚在 Check-out 狀態等資訊，本系統提供查詢功能，讓使用者快速查詢到所需資訊，圖 27。

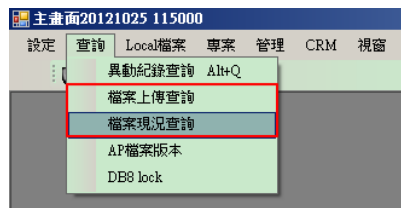


圖 27 檔案版本控管服務查詢選單

檔案現況查詢，主要為查詢目前 Check-out 之專案，以利使用者瞭解尚未完成之專案資料及其版本與 Check-out 之使用者，圖 28。

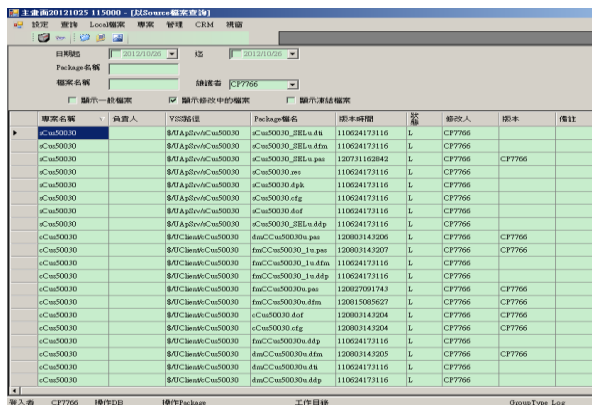


圖 28 檔案版本控管服務檔案現況查詢畫面

檔案上傳查詢，主要提供使用者查詢已 Check-in 之專案資訊，以利使用者瞭解專案 Check-in 歷史記錄及其異動內容，圖 29。

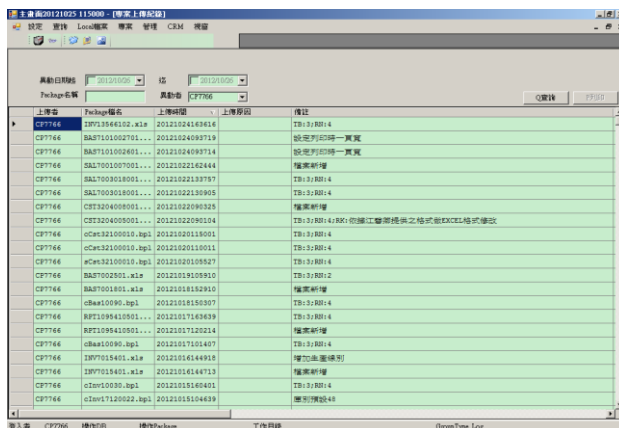


圖 29 檔案版本控管服務檔案上傳查詢畫面

(4) 比對

為使使用者能快速比對專案目錄下檔案之歷史版本，比對功能有兩種，一是將目前 Check-out 後之檔案與其目前最新檔案版本作比對，可於使用者端主畫面右方檔案利用滑鼠右鍵選單選擇「顯示目前版本差異」，主要功能為比對工作目錄與最後上傳檔案版本之差異比對，圖 30~圖 31。

另一則是顯示該檔案之歷史版本後，再對其任一歷史版本作比對。因檔案歷史版本可能過多，可於使用者端主畫面右方檔案利用滑鼠右鍵選單選擇「顯示 History」，開啟歷史版本畫面，主要功能為顯示該檔案之歷史版本與其被使用者的異動動作資訊，以供使用者用透過 Web service 將雲端檔案儲存服務上之專案檔案進行不同版本之比對差異。在進行不同版本間之比對差異。可供比對版本操作類型需為 W(Check-in)。因操作類型 W 代表已將專案檔案透過 Web service 存入雲端檔案儲存服務，並對原專案複製更名後產生一份專案新版本的異動資訊，而比對功能則是比對已存在的各種版本差異，對於顯示該檔案之歷史版本的異動行為，Check-out、UndoCheck-out 等異動行為之專案檔案進行版本比對較無意義，圖 32~圖 33。

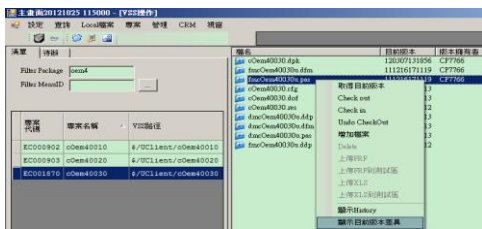


圖 30 檔案版本控管服務顯示版本差異畫面

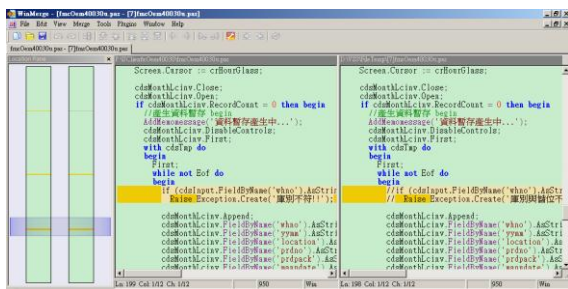


圖 31 版本差異比對畫面

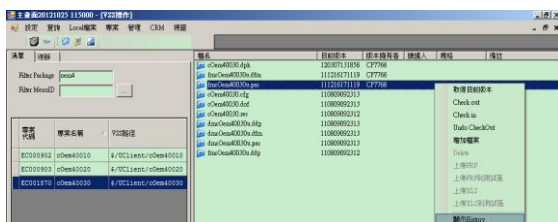


圖 32 檔案版本控管服務顯示 History 畫面



圖 33 檔案版本控管服務之歷史版本比對

### 4.3 FVCS 實作小結

本系統已建構並且實作於台灣某食品股份有限公司資訊部門一年多餘，該公司一年多來有關 UClient 類別已產生 25,407 個檔案、2,124 個專案目錄，共計 547MB 容量大小，UApSrv 類別已產生 10,557 個檔案、1,291 個專案目錄，共計 20MB 容量大小，資料庫已產生 249,824 筆資料，共計 30MB。

每日上傳檔案數約 12~15 個不等，依照此成長速度來看，保守估計在三年內將可達到雲端檔案儲存服務免費容量的上限（以 Dropbox 為例，免費空間為 2GB），屆時則需要考慮購買雲端檔案儲存服務的空間或是自行構建私有雲端檔案儲存服務。就企業成本考量方面，不需負擔自行構建私有雲端檔案儲存服務，改由使用公有雲端檔案儲存服務，將可使企業節省約三年的硬體構建與維護費。

而除了節省硬體成本外，就該部門同仁也因檔案版本控管系統改善了檔案控管、區域及便利等狀況之工作效率。檔案版本不再如未使用檔案版本控管前的凌亂，因有規則性的管控，在編輯修改軟體時不會因為檔案版本不同而造成日後軟體開發時功能的遺失，甚至同仁不在上班時間與上班環境下，也可協助公司處理事務，提升使用者便利性。

本章將就研究結果與版本控管系統 VSS 比較發現，列出不同之優點說明，如表 8：

表 8 本研究與版本控管系統 VSS 比較

	版本控管系統	本研究 FVCS
空間	視安裝主機容量	視雲端儲存空間
備援	需另外建構	有支援
資料庫負載	高	低
區域	區域	廣域
資料安全	有支援	有支援
便利	需安裝 VSS 於伺服器，無法隨時隨地修改	不需安裝於伺服器，可隨時隨地修改
成本考量	版權付費	免費

(1)相對於空間性，本研究參考現有雲端檔案儲存服務的基礎來架構檔案版本控管系統，本系統與其它的版本管控系統皆是採用主從式架構，也就是 Client/Server 架構，但其它的

版本管控系統的儲存層大都是採用 Server 主機的硬碟空間或是將檔案存進資料庫裡，備份檔案資料時，也是儲存於 Server 主機，這會造成 Server 主機的容量日趨縮小。儲存層於雲端檔案儲存服務的資料，因資料本身就是存在於雲端，雲端又有備援的機制，也就是說，不用一直購建硬體設備來做儲存容量的擴增，只要利用雲端檔案儲存服務為企業提供空間如 Dropbox，就可以做到儲存檔案與備援檔案的功能，相對於其硬體成本會較少於企業自行構建硬體設備來的低。

- (2)相對於備援性，雲端儲存服務具有即時同步功能，只要偵測到有新增或是修改、刪除檔案等異動，會立即同步該檔案資訊內容。除了檔案同步功能外，雲端儲存服務還具有備份與還原機制，對於檔案版本控管系統來說，等於是具備雙重備援之機制。
- (3)相對於資料庫負載性，將檔案儲存於實體硬碟與檔案至轉換成二進位儲存於資料庫來說，在存取時，不必經過轉換格式的手續，也不會造成資料庫的負荷與容量的成長。一旦資料庫有異常，造成資料無法讀取，那麼儲存於資料庫的檔案資料也會一併毀損無法救回，雖說有備援機制可供挽回檔案的功能，但不知資料庫毀壞的程度如何，也就無法得知有哪些檔案資料是已經毀損的。本研究將檔案儲存於實體硬碟，存取時不需再透過特定轉換程式，直接取得檔案資料，讓資料庫單純只儲存使用者的異動行為，做到檔案與資料分開，不會造成資料庫毀損，最為重要的程式碼無法救回的窘境。
- (4)相對於區域性，一般的檔案控制系統，僅能建構於區域網路下，也就是在企業內部工作環境下，才可以編輯修改檔案，且記錄使用者各種異動處理的狀態與資料，若是處於非企業內部工作環境下，則必須透過遠端連線回電腦並操控電腦進行編輯修改，可能在連線過程中，因網路頻寬或速度問題，無法快速的修改編輯檔案，因而造成企業系統的異常。本研究透過主從式架構與 Server 端溝通，可達成廣域網路存取檔案，只在上傳下載檔案時需透過網路傳遞資訊，其修改編輯時的處理都在本機 Client 端，不再因為遠端連線時頻寬問題，往往開啟一個圖片或是檔案時，在經過數秒或是數分鐘，遠端桌面才執行，造成時間與效率的浪費。

(5)相對於資料安全性，利用企業內部ERP系統之登入帳號密碼來進行驗證，並且無法對現有檔案進行修改或刪除，僅能對現有檔案另存新版本後編輯與上傳，使用者的異動行為，清楚的被記錄下來，增強對檔案的保護與控管。在檔案版本控管部份，因有使用者異動的記錄，對於使用者對檔案的版本做比對或是編輯能更有效的知曉修改歷程。

(6)相對於便利性，使用者因有雲端儲存服務，提供任何電腦都能連接，並透過檔案版本控管機制，讓資料異動能被需要的人查詢利用，效能與便利性優於公司內部一般人員自行做檔案版本控管或是透過檔案版本控管系統的方式，也可藉此取代隨身碟當作儲存媒介，禁止隨身碟使用，落實企業資料安全控管。

(7)相對於成本考量性，因使用的是現有的雲端儲存服務，目前現有的雲端儲存服務有提供不小的免費空間(以 Dropbox 為例，免費空間 2GB)，對於軟體開發程式碼而言，已是足夠儲存與備援，相對於企業與個人來說，不需再另建硬體設備來擴充儲存空間，節省最大開銷成本的支出。

## 5. 結論與建議

本研究以植基於雲端檔案儲存服務之檔案版本控管架構為研究範圍，開發出 FVCS 系統，並實作於台灣某食品股份有限公司資訊部門，在本章節中，將說明於該公司在實作後的結論與貢獻，並提供未來進行相關研究的建議方向。

### 5.1 結論與貢獻

在軟體開發的階段中，版本控管是一件很重要的事情，由於開發成員的異動或是軟體規格沒有更新與開發成員自顧自地建立程式原始碼的備份機制，會帶來團隊在軟體開發時檔案版本的錯誤與混亂，雖然利用版本控管系統可以協助軟體開發團隊以管理、記錄、備份每次設計的成果，但僅限於區域網路的實行，無法在非上班時間與上班環境下執行版本控管，本研究已完成植基於雲端檔案儲存服務之檔案版本控管架構，採用主從式架構，區分為三個層次，分別為應用層、控制層、儲存層，利用使用者前端互動介面作為應用層，將使用者的申請需求透過控制層來與資料庫溝通並回饋使用者資訊如各式查詢報表、取得檔案版本比對



等，或是使用者申請將修改後的專案版本上傳，經由控制層將新專案版本新增或更新至儲存層，以這三層的設計架構來達到大空間、雙備援、跨區域、極便利與低成本等優點。

本研究 FVCS 系統成功實作於台灣某食品股份有限公司資訊部門一年多，以該部門使用 FVCS 系統至今，已如本研究設計時所預期，以不自行負擔硬體構建與維護為前提下，可以順利於非上班時間與非上班環境下，編輯查詢檔案版本並上傳於雲端儲存服務 Dropbox，大大的加強了該部門的工作效率與便利性。

## 5.2 建議與未來研究

在系統實作與運作後發現幾點在未來可以進行強化與優化：

- (1)可在控制主機架構防毒服務，結合防毒軟體可以確保上傳檔案不受病毒威脅，並能降低使用者透過本系統增加病毒感染率。
- (2)雲端儲存服務本身就具備有備份機制，但自行建置的雲端儲存服務並無備份機制，所以在建置私有雲時，除了能提高儲存空間與安全性外，需再加入備援機制。
- (3)比對軟體可供比對的檔案有限制，目前僅能比對文字檔，對於 Word、Excel 等特殊軟體，無法進行比對，以提高系統可用度。
- (4)可設計成網頁方式以提升使用度，不用下載 Client 端軟體，可達到走到哪裡服務到哪裡。

## 參考文獻

- [1]王唯達，“SaaS 雲端運算之分散式異質資源整合與研究：以台灣文學館電子資源整合查詢為例”，碩士論文，臺南大學資訊工程學系，2010。
- [2]王國良，“雲端與傳統運算架構之效義比較分析：以海軍傳艦修護資訊系統為例”，碩士論文，銘傳大學資訊管理學系，2011。
- [3]李宗南，“雲端算不算”，科學發展專題報導，第 461 期，第 38-41 頁，2011。
- [4]周芋玲，“雲端技術應用於電腦題庫系統之研究”，碩士論文，世新大學新聞傳播學系，2011。
- [5]林秉毅，“具版本管理之備份機制的設計與實作—以 XML 為基礎”，碩士論文，朝陽科技

大學資訊管理系，臺中，2006。

- [6]洪菁憶，“循序探勘在軟體版本控制上的應用”，碩士論文，國立中央大學資訊管理研究所，桃園，2008。
- [7]修丕承、張育銘、郭大維，“未來行動裝置與其軟體系統”，NCP Newsletter，第 19 期，第 2-10 頁，2010。
- [8]陳兆宏，“雲端運算離線之備援與代理機制”，碩士論文，大同大學資訊工程研究所，2012。
- [9]虞為、陳俊鵬，“基於本體的雲計算文獻管理模式研究”，情報雜誌，第 30 卷，第 2 期，第 147-151 頁，2011。
- [10]廖啟志，“運用雲端儲存技術建構企業私有儲存雲”，碩士論文，亞洲大學資訊管理學系，2010。
- [11]維基百科，(CVS)Concurrent Versions System，擷取於維基百科，2012。
- [12]維基百科，(VSS) Visual SourceSafe，擷取於維基百科，2012。
- [13]維基百科，Microsoft SQL Server 2008 Express，擷取於維基百科，2012。
- [14]蔡煥麟，版本控制系統的基礎觀念，擷取於 <http://www.dotblogs.com.tw/huanlin/archive/2008/04/23/3198.aspx>，2012。
- [15]Chen, A., Chou, E., Wong, J., Yao, A. Y., Zhang, Q., Zhang, S., “CVS Search: Searching through source code using CVS comments,” IEEE International Conference Software Maintenance (ICSM), pp. 364-374, 2001.
- [16]Rochkind, Marc J., “The Source Code Control System,” IEEE Transaction on Software Engineering, pp. 364-370, 1975.
- [17]Tichy, Walter F., “RCS-A System for version Control,” Software-Practice & Experience, Volume 15, Issue 7, pp. 637-654, 1985.
- [18]Wesfechtel B, Munch B P and Conradi R., “A Layered Architecture for Uniform Version Management,” IEEE Transactions on Software Engineering, pp. 1111-1133, 2001.